Perception monoculaire de l'environnement pour les systèmes de transport intelligents

Yann Dumortier

ED n°431 : Information, communication, modélisation et simulation

T H È S E

pour obtenir le grade de

DOCTEUR DE L'ÉCOLE NATIONALE SUPÉRIEURE DES MINES DE PARIS

Spécialité "Informatique temps-réel, robotique, automatique"

présentée et soutenue publiquement par
Yann DUMORTIER

le 16 octobre 2009

PERCEPTION MONOCULAIRE DE L'ENVIRONNEMENT POUR LES SYSTÈMES DE TRANSPORT INTELLIGENTS

Directeur de thèse : Isabelle HERLIN
Co-encadrant : André DUCROT

Jury

M. Gérard MEDIONI	Président
M. Thierry VIEVILLE	Rapporteur
M. Abder KOUKAM	Rapporteur
M. Benoist FLEURY	Examinateur
M. Arnaud de LA FORTELLE	Examinateur

Remerciements

Je tiens à saluer ici les personnes qui, de près ou de loin, ont contribué à la concrétisation de ce travail. Ces remerciements sont rédigés dans un moment de relâchement intellectuel, après trois années et demie d'investissement total, par certains aspects coupé du monde extérieur. Aussi mes pensées se dirigent-elles tout d'abord vers ma famille et mes proches, pour leur patience et leur compréhension tout au long de cette étape décisive de ma vie.

Je souhaite également remercier les personnes qui m'ont offert de travailler dans un environnement stimulant, sur un sujet d'étude aussi motivant que les systèmes de transport intelligents : Michel Parent, Directeur du projet IMARA, qui m'a accueilli au sein de son équipe, ainsi que Claude Laurgeau et Arnaud de La Fortelle, tous deux Directeurs du centre de robotique de l'Ecole des Mines de Paris où j'ai eu plaisir à échanger avec l'ensemble des chercheurs présents durant ma thèse.

Bien sûr, mes remerciements vont conjointement à Isabelle Herlin et André Ducrot, mes Directeurs de thèse, grâce auxquels j'ai pu mener à bien ce projet. Leur tâche n'a pas toujours été simple, mais leur complémentarité et leur disponibilité m'a permis d'appréhender, chaque jour d'avantage, les exigences du métier de chercheur.

Je remercie Thierry Vieville, Directeur de recherche à l'INRIA, ainsi qu'Abder Koukam, Professeur à l'Université de Technologie de Belfort-Montbéliard, de m'avoir fait l'honneur d'accepter de juger ce mémoire.

Je tiens en outre à exprimer ma profonde gratitude à Gérard Medioni, Professeur à l'Université de Californie du sud et président du jury, pour l'ensemble des conseils prodigués tandis que je visitais son laboratoire au cours de mon doctorat.

Enfin, mon travail n'aurait pu être pleinement évalué sans l'avis d'un acteur reconnu de l'industrie des systèmes d'aide à la conduite, Benoist Fleury, du groupe Valéo, que je remercie pour sa participation au jury de thèse.

Pour clore ces remerciements, je tiens à adresser toute ma reconnaissance à des personnes qui, dans le cadre du travail et parfois en dehors, m'ont soutenu tout au long de ces années. Je pense à Chantal Chazelas, Konaly Sar et Christine Vignaud, assistantes de la JRU laRA, ainsi qu'à la *Dream Team* du Service des Ressources Humaines du centre INRIA Rocquencourt : Claire Alexandre, Fatima Ayad, Noelle Bourgeois, Myriam Brettes, Myriam Chaazal, Catherine Chaix, Françoise Feneck, Martine Girardot, Alix Michon et Catherine Verhaeghe. Qui a dit "que des femmes" ?

Table des matières

L'évolution des transports, au cours des dernières décennies, témoigne d'une volonté continue de réduire les contraintes associées à la notion de déplacement. Dans ce but, une part importante des efforts engagés a pour objectif de raccourcir la durée des trajets, essentiellement grâce à l'amélioration des infrastructures et la diversification des modes de transport. La multiplicité modale, censée répondre aux différents besoins des usagers, n'a cependant pas suffi à stopper l'essor de l'automobile au sein des agglomérations. La voiture individuelle y est ainsi progressivement devenue la principale source de nuisances sociétales, environnementales et sanitaires.

Les solutions étudiées pour remédier à cette situation reposent principalement sur la responsabilité du facteur humain. Elles proposent donc notamment de remplacer l'automobile par des systèmes de transport autonomes. L'automatisation des véhicules, progressivement mise en place par la démocratisation des systèmes d'aide à la conduite (ADAS[1]), nécessite dès lors le développement de modules de perception de l'environnement, qui analysent et traitent l'information acquise à partir d'un ou plusieurs capteurs. Avec l'explosion des capacités de calcul des systèmes embarqués, la caméra est ainsi devenue l'un des capteurs les plus utilisés, tant pour la richesse de l'information contenue dans une séquence d'images, que pour son faible coût et son encombrement limité. Aussi, la vision artificielle en robotique mobile, notamment dans le cadre des transports intelligents, est un sujet largement traité dans la littérature.

Dès la fin des années quatre-vingt, les premiers systèmes de perception de l'environnement appliqués aux véhicules routiers permettaient déjà de segmenter la route dans une image suffisamment structurée. Seule l'information 2-D contenu dans le plan focal était alors exploitée grâce à l'emploi d'une architecture matérielle spécifique. Aujourd'hui, la plupart des approches modernes tentent de retrouver l'information de profondeur à l'aide de contraintes géométriques sur plusieurs acquisitions. L'utilisation de ces contraintes implique cependant de travailler sur une partie restreinte des points de l'image pour une exécution temps-réel. En parallèle, le développement des méthodes de reconnaissance par apprentissage a offert la possibilité d'une analyse sémantique plus fine de la scène, mais la complétude des bases d'apprentissage correspondantes, et donc la robustesse de ces systèmes, ne peuvent être garantie.

---

[1] ***A**dvanced **D**riving **A**ssistance **S**ystem.*

Les travaux présentés dans le cadre de cette thèse apportent une solution originale aux problèmes de perception visuelle pour la conduite automatisée, à travers une approche monoculaire fondée sur l'étude des contraintes géométriques précédemment énoncées. Ces dernières sont appliquées sur le résultat filtré de la mise en correspondance, dense, des points de deux images consécutives. Le choix d'une solution matérielle et logiciel appropriée, grâce à l'emploi de la technologie Nvidia CUDA, permet de pallier la complexité algorithmique, d'une partie du processus, par sa parallélisation. Le champ de vecteurs vitesse calculé est filtré par Tensor Voting*, selon un critère de continuité sur l'espace joint des positions et déplacements. Le modèle dynamique de l'espace navigable peut ainsi être estimé de façon plus robuste et rapide qu'à l'aide du flot brut. La segmentation du domaine libre pour circuler est finalement définie par comparaison de ce modèle avec le mouvement image, tandis que l'étude de la parallaxe permet de discriminer les obstacles mobiles. Il est ainsi possible d'obtenir une représentation suffisante de l'environnement pour permettre au module de plannification de trajectoire d'assurer la sécurité du véhicule autonome.*

La suite du document est organisée de la façon suivante. Une première partie, introductive, détaille la problématique liée à l'organisation des transports, jusqu'au développement de systèmes alternatifs à l'automobile. Les parties deux et trois traitent ensuite de la segmentation du mouvement image et des contraintes géométriques qui sont appliquées au champ de déplacement afin de percevoir numériquement l'environnement. La dernière partie conclut enfin sur l'intégration de l'approche présentée, à l'aide notamment d'une carte graphique pour en accélérer le traitement.

Première partie

Contexte de l'étude

La popularisation de l'automobile au début du vingtième siècle, notamment au sein des grandes agglomérations, a permis d'améliorer la mobilité de ses usagers. Ce nouveau mode de locomotion, individuel et privé, répond alors à l'insuffisance du réseau de transport collectif de l'époque. Rapidement, le véhicule particulier se révèle néanmoins être la source de nombreux problèmes sociétaux, sanitaires et environnementaux, dont la responsabilité est imputée aux conducteurs. Pour y répondre, différents scénarios visant à remplacer l'automobile par un mode de transport intelligent, individuel et public, sont à l'étude depuis le milieu des années quatre-vingt-dix et la naissance du projet CyberCar. *L'automatisation de ces véhicules nécessite de concevoir un module de perception, capable d'analyser l'environnement à l'aide des données retournées par les différents capteurs embarqués. Le choix de ces derniers est donc un élément critique, tant pour la pertinence des informations acquises que pour le coût du système développé.*

Après avoir établi un bilan des méfaits de l'automobile sur l'environnement et la santé, le premier chapitre poursuit sur le développement des systèmes d'aide à la conduite, jusqu'à la conception d'un nouveau mode de transport intelligent. Un second chapitre étudie ensuite les besoins inhérents au processus de perception pour la conduite automatisée, et propose un modèle efficace de l'environnement ainsi que le capteur adéquat pour construire ce dernier.

Chapitre 1

Les systèmes de transport intelligents

L'organisation du réseau de transport moderne, qui encourage l'utilisation de l'automobile, est à l'origine des problèmes sanitaires, sociétaux et environnementaux, imputés aux véhicules particuliers. La responsabilité avérée du facteur humain, à ce sujet, conduit donc les constructeurs automobiles à améliorer sans cesse les systèmes d'aide à la conduite. Inéluctablement, la voiture s'automatise, et la vision d'un nouveau mode de transport public, individuel, se concrétise.

Après avoir établi, dans une première section, un état des lieux des systèmes de transport urbains et péri-urbains, ainsi que le rôle de la voiture en leur sein, la suite du chapitre présente le concept de véhicule automatisé comme solution aux problèmes induits par l'automobile.

1.1 Les transports aujourd'hui

1.1.1 Intermodalité et mobilité

On définit la mobilité d'une population en fonction du temps accordé quotidiennement aux transports. La croissance du volume des déplacements est étroitement liée à l'augmentation des vitesses atteintes par les nouveaux moyens de locomotion ainsi qu'à l'amélioration des infrastructures disponibles. C'est pourquoi, dans la seconde moitié du XXème siècle, avec le développement urbain autour de l'automobile, la périphérie des villes des pays industrialisés est devenue plus attractive. De multiples facteurs, comme la péri-urbanisation qui s'en est suivie et l'allongement des distances de déplacement quotidien, ont alors contribué à faire

de la voiture individuelle le moyen de transport dominant dans la majorité des pays européens.

Face à la croissance régulière du parc automobile et aux nuisances associées, les autorités gouvernementales ont mis en place des politiques de déplacement urbain visant à équilibrer l'utilisation des différents moyens de transport. Le résultat des ces politiques a été la création d'un maillage articulé autour de pôles d'échange multi-modaux, des lieux au sein desquels un ensemble d'installations permet aux usagers des transports d'accéder à différents modes de déplacement. Ces interfaces sont agencées dans l'espace urbain et péri-urbain de manière à répondre au mieux aux besoins des utilisateurs dans la gestion de leur mobilité quotidienne :

- les aéroports et gares centrales connectent les réseaux intra et extra-urbains ;
- des parkings d'échange permettent le relais entre le réseau routier périphérique et les réseaux de transport collectif ;
- enfin, des interfaces inter-modales urbaines servent de passerelles entre les différents réseaux de transport collectif.

L'usager peut ainsi optimiser son déplacement par le choix, à chaque étape de son parcours, du mode de transport le mieux adapté.

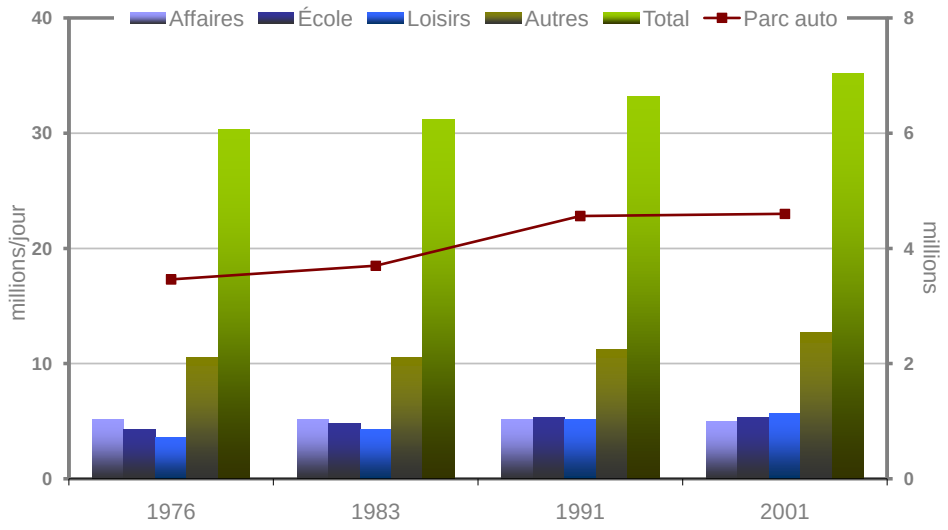


FIG. 1.1 – Évolution comparative du volume de déplacements et du parc automobile en Île de France.

La flexibilité de ce modèle multi-modal permet d'intégrer facilement tout nouveau moyen de locomotion. La mise en place du système *Vélib'* au sein de la ville de Paris, en juillet 2007, est certainement l'exemple le plus connu. Ce mode de transport public et individuel, basé sur la location de vélos en libre service, connaît depuis un succès grandissant, malgré le manque d'infrastructures spécifiques et de nombreux heurts entre cyclistes et automobilistes. En répondant à une réelle demande sociale, le système *Vélib'* a démontré son utilité en complément des réseaux de transports urbains pré-existants. Cependant, les problèmes répétés de congestion du trafic routier témoignent de la persistance d'un déséquilibre important, ayant pour origine l'utilisation massive de la voiture individuelle en agglomération. Les chiffres publiés par la dernière Enquête Globale de Transports (EGT'01) confirment ce constat, en indiquant une augmentation du parc automobile ainsi que de la mobilité des Franciliens ces 20 dernières années (Fig. 1.1). Au niveau national et européen, la tendance semble identique avec un taux de croissance important du nombre des nouvelles immatriculations, respectivement de 3.2 et 0.9 % entre 2005 et 2006[1]. En outre, certains experts estiment que le nombre d'automobiles dans le monde pourrait doubler dans les vingt prochaines années, compte tenu de la demande croissante des pays émergents.

1.1.2 L'automobile comme source de problèmes

Problèmes sociétaux

Au succès retentissant de l'automobile a rapidement suivi une dégradation des conditions de circulation. Près de 45 % de la durée du trajet parcouru par les usagers du réseau péri-urbain, aux heures de pointe de la matinée[2], s'effectue dans un trafic saturé, soit à moins de 30 km/h. Les performances du conducteur en terme d'usage de l'espace sont en cause : il lui faut en moyenne une largeur de chaussée de 3,5 mètres, pour un véhicule n'excédant que rarement 2 mètres de large, et une distance de sécurité égale à celle parcourue en deux secondes (soit deux fois le temps de réaction humain). Ainsi, le débit d'une voie rapide ne dépasse pas 2500 passagers à l'heure quand, pour le même espace, les trains, métros ou tramways sont susceptibles de transporter plus de 30000 passagers à l'heure.

La gestion de l'espace urbain doit, de plus, faire face à la demande croissante de places de parking. Un véhicule automobile reste en moyenne 95% du temps en

[1] Source : Forum International des Transports 2008.
[2] 6 heures - 10 heures.

stationnement, où il occupe environ 10 mètres carrés en bord de voirie. En tentant de répondre aux besoins des usagers par la construction de parkings souterrains, les politiques d'urbanisation ont eu un effet aggravant, en augmentant le parc automobile et en favorisant ainsi l'engorgement des centres-villes.

Ces difficultés, auxquelles il faut ajouter le bruit des véhicules et l'odeur des gaz d'échappement,affectent aussi bien les conducteurs que les résidents des agglomérations, occasionnant stress et agressivité. Enfin, la taxation des infrastructures de stationnement, le coût du carburant et les difficultés d'accès à l'automobile pour certaines catégories sociales de la population (personnes âgées, invalides, etc.) sont autant de facteurs discriminants, au sein d'une société favorisant l'utilisation de véhicules particuliers.

Problèmes environnementaux

L'automobile est également une source importante de pollution, notamment par l'émission de CO_2, gaz à effet de serre. Alors qu'entre 1980 et 1997, le trafic automobile a progressé de 55 %, la consommation unitaire moyenne des véhicules en carburant (à laquelle sont directement liées les émissions de CO_2) a baissé seulement de 18 %[3]. Les émissions totales de CO_2 ont donc sensiblement augmenté durant la même période, malgré l'amélioration du rendement énergétique unitaire. En France, on estime que pour l'année 2007, 25 % des rejets de CO_2 sont à imputer aux transports routiers.

En outre, quelles que soient les solutions techniques adoptées (carburant vert, véhicule électrique, etc.), les performances énergétiques de l'automobile sont étroitement liées au type de conduite adopté. Ainsi, la congestion du réseau routier, qui ne permet pas le maintien d'une allure régulière, est un facteur aggravant des problèmes environnementaux.

Problèmes sécuritaires et sanitaires

En 2006, le nombre d'accidents de la route en Europe a diminué de 1.5 % par rapport à l'année 2005, avec néanmoins plus de 1.2 millions de sinistres, 1.6 millions de blessés (-2.5%) et 39 000 tués (-3.7 %). Dans le même temps, la politique répressive menée en France, pour réduire la vitesse moyenne des automobilistes, a permis une baisse significative de 11.5 % du nombre des tués[4]. Cependant, les

[3]Source : Club d'Ingénierie Prospective Énergie et Environnement (2001).

[4]Source : Forum International des Transports 2008.

avancées en matière de sécurité routière et les progrès réalisés en terme d'infrastructures sont limités par le facteur humain, mis en cause dans la grande majorité des accidents. Le temps de réponse du conducteur et ses réactions inadéquates, voir accidentogènes dans 30 % des situations, font de l'automobile le mode de transport le moins sûr actuellement.

Il est nécessaire d'ajouter à ce constat, les victimes indirectes de l'automobile. L'étude Erpurs, de l'Observatoire Régional de la Santé, a en effet pu mettre en évidence des corrélations à court terme entre pollution de l'air, absentéisme professionnel, urgences pédiatriques et mortalité.

1.1.3 Une solution de transition : le véhicule intelligent

Face aux attentes des usagers de la route en matière de sûreté, la sophistication des organes de sécurité est devenue une composante majeure du processus de modernisation de l'automobile. Les dispositifs de sécurité passifs, tels que la ceinture de sécurité, les renforts latéraux ou l'airbag, ont été les premiers à être intégrés dans les véhicules particuliers. Ils ont pour rôle de limiter les conséquences graves d'un accident. D'autres systèmes, véritables relais à la vigilance du conducteur, sont depuis capables d'aider, d'avertir, voire d'assister l'usager en cas de besoin. Ainsi, la projection holographique des informations de navigation sur le pare-brise, la vision nocturne par caméra thermique ou la détection des panneaux signalétiques, sont trois exemples de dispositifs peu invasifs d'aide à la conduite. A l'inverse, d'autres systèmes ont la capacité d'agir directement sur la commande du véhicule et donc de remplacer partiellement le conducteur :

- l'Assistance au Freinage d'Urgence (AFU) peut accentuer la pression de freinage en fonction de la vitesse d'enfoncement de la pédale de frein ;
- le correcteur électronique de trajectoire (ESP[5]) est capable de corriger la trajectoire du véhicule en agissant, indépendamment pour chaque roue, sur le système de freinage ;
- le contrôle adaptatif de vitesse (ACC[6]) régule la distance séparant l'automobile du véhicule précédent, à l'aide d'un radar placé à l'avant de la voiture. Il peut actionner, le cas échéant, le système de freinage ;
- enfin, le stationnement automatique permet, sur certains modèles tels que la Citroën C3 City Park ou la Toyota Prius II, de se garer sans aucune intervention humaine.

[5] Electronic Stability Program.
[6] Adaptive Cruise Control.

Il est probable que les constructeurs automobiles proposeront encore longtemps des véhicules intégrant des dispositifs de plus en plus sophistiqués d'aide à la conduite et à la navigation, tout en maintenant le conducteur comme composante majoritaire de la boucle de contrôle. Néanmoins, on estime que pour améliorer de façon déterminante la sécurité routière, ces mêmes manufacturiers devront peu à peu abandonner le contrôle des véhicules à l'électronique embarquée.

1.2 La voiture de demain

1.2.1 Le véhicule automatique

Selon une étude anglaise, il apparaît que le transfert modal vers les transports collectifs de seulement 5 % des automobilistes londoniens, permettrait, aux usagers de la route restant, de gagner en moyenne 4 minutes par trajet. Les utilisateurs habituels de bus gagneraient de même 5 minutes par trajet. En revanche, les 5 % de personnes ayant changé de mode de transport accuseraient dès lors un retard de 16 minutes. Le résultat de cette étude est double. D'une part, il montre combien une faible baisse du nombre des véhicules en circulation peut avoir un impact significatif sur la congestion du trafic ; d'autre part, il rappelle les raisons du succès de l'automobile : un moyen de locomotion souple et rapide qui permet un gain de temps au conducteur dans la majorité des situations. La voiture ne semble donc pas prête à disparaître du paysage urbain et péri-urbain, même si pour trouver sa juste place au sein des différents réseaux de transports, elle doit radicalement évoluer.

Les principaux pays industrialisés ont pris conscience de la gravité de la situation et de nombreux projets scientifiques, notamment européens, ont ainsi vu le jour sur ce thème ces vingt dernières années. Parmi eux, certains programmes préconisent l'automatisation complète du parc automobile. La suppression du facteur humain aurait pour conséquences directes :

- d'augmenter le débit des voies de circulation en diminuant la distance de sécurité inter-véhicule, imposée jusqu'alors par le temps de réaction humain ;
- de supprimer les conduites à risques ;
- de permettre l'accès à la voiture au plus grand nombre (personnes âgées, invalides, sans permis, etc.).

Parallèlement, la mise en place d'un système de partage dans le temps des véhicules, libérerait les centre-villes d'une part importante de leurs aires de stationnement. L'automobile fonctionnerait alors sur le principe d'un taxi automatisé, à

disposition sur demande et dans toute la zone desservie. De précédentes recherches ont montré qu'un tel scénario, grâce à l'automatisation des carrefours et à la décongestion du trafic, augmenterait la vitesse moyenne des automobiles jusqu'à 44 km/h, porte-à-porte. On estime ainsi qu'une voiture en libre-service remplacerait près d'une quinzaine de véhicules particuliers, et améliorerait la mobilité des citadins en supprimant toute dépendance envers un mode de locomotion particulier : au cours de la journée, les utilisateurs adapteraient plus facilement leur moyen de transport en fonction de leur besoin effectif. Les effets de la réduction du parc automobile sur le volume total des émissions polluantes auraient, de plus, un impact environnemental positif.

1.2.2 Schéma fonctionnel

La conduite automatisée est possible moyennant une plateforme mobile, instrumentée et communicante. L'instrumentation embarquée, qui regroupe l'ensemble des capteurs à disposition, doit permettre au véhicule de se localiser et de progresser au sein d'un environnement dynamique. La partie communicante, quant à elle, se compose d'une interface utilisateur et de moyens d'échange avec les infrastructures environnantes ou avec d'autres véhicules automatisés. L'usager a pour seul rôle la saisie d'une destination, via une base de données géo-référencée de type cartographie électronique, tandis que les éléments de communication avec les systèmes distants permettent au véhicule intelligent d'accéder aux capteurs de ces derniers.

L'automatisation du déplacement requiert également l'intégration logique d'un processus de perception multi-échelles. L'itinéraire doit d'abord être défini au niveau macroscopique par une liste de points de passage sur un modèle statique du monde. Cette liste est en permanence remise à jour pour palier à toute dérive du véhicule hors du chemin prédéfini. A l'échelle locale, la trajectoire du véhicule est ensuite calculée dynamiquement, entre la position courante et la position du prochain point de passage. Elle doit tenir compte de l'environnement immédiat du véhicule, incluant les obstacles mobiles pouvant s'y trouver. Enfin, la partie contrôle-commande agit alors sur les actionneurs pour suivre au plus près la trajectoire pré-estimée.

Le module de perception occupe donc la place centrale au sein de l'organigramme fonctionnel des véhicules autonomes (Fig. 1.2). Il reçoit, analyse et traite les données provenant des différents capteurs, installés afin de renseigner l'ensemble des modules de la couche logique sur l'état du véhicule et son environ-

nement.

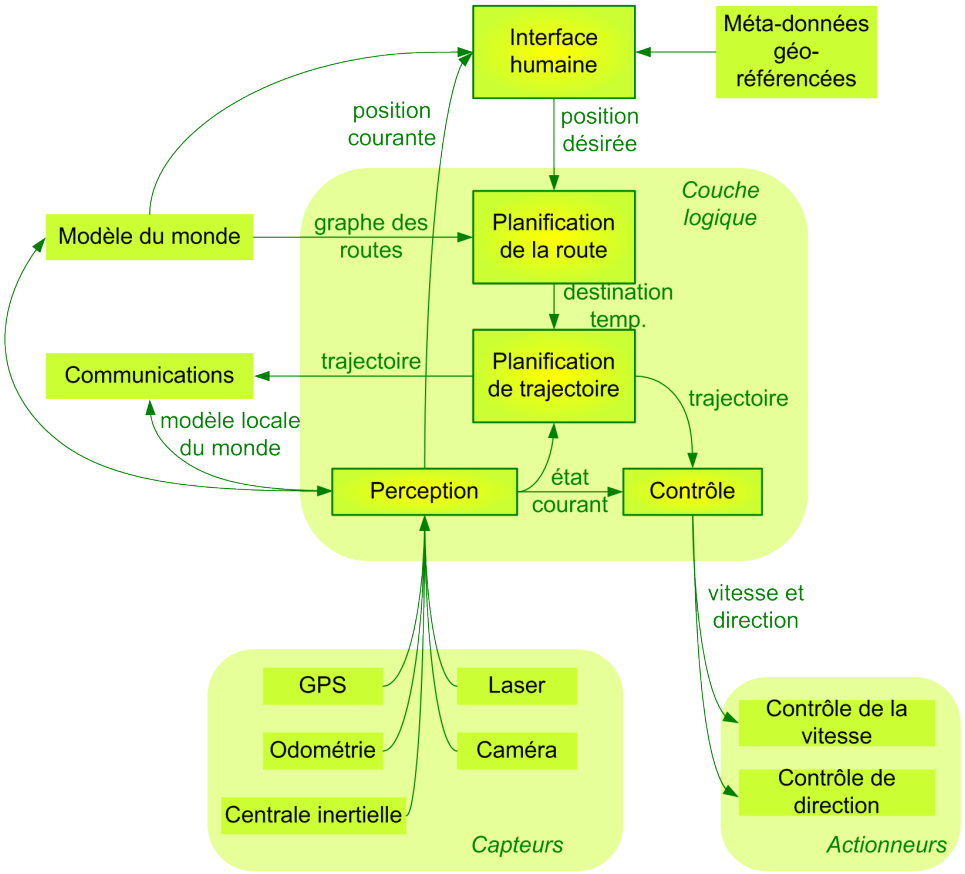


FIG. 1.2 – Schéma fonctionnel d'un système automatisé de type Cycab.

Chapitre 2

Perception de l'environnement pour la conduite automatisée

La perception de l'environnement est le processus de recueil et de traitement de l'information sensorielle permettant de définir le milieu occupé. Toutefois, dans le cadre d'une activité spécifique, telle que la conduite, tous les détails de l'environnement ne correspondent pas nécessairement à une information utile. Aussi est-il essentiel d'appréhender correctement les besoins relatifs au contexte, afin de déterminer une représentation optimale du milieu. La suite du chapitre s'attache donc, dans une premier section, à définir un modèle minimal de l'environnement pour la conduite automatisée. Une seconde section établit ensuite le type de capteur adéquat pour construite ce modèle.

2.1 Définitions

2.1.1 Navigation et modèle minimal étendu de l'environnement

La navigation est un processus qui regroupe l'ensemble des techniques assurant la localisation et le calcul de routes dans un référentiel unique. Elle nécessite un système de positionnement associé à un modèle du monde, pour permettre d'évoluer sans risque de collision avec les différents éléments de l'environnement au cours du trajet parcouru. A ce jour et en l'absence de solution alternative fiable et accessible, la géo-localisation par satellite est la technologie de référencement géographique la plus communément déployée dans les domaines civil et militaire.

Le système américain GPS[1], seul dispositif actuellement opérationnel, couvre l'intégralité du globe avec une constellation de 24 satellites orbitant à 20200 kilomètres d'altitude. Il n'offre cependant ni la précision, ni la robustesse suffisante pour concevoir un véhicule automatisé naviguant grâce à sa seule utilisation. L'erreur de positionnement horizontal peut en effet atteindre 15 mètres et la qualité de réception des signaux satellitaires dépend fortement de la configuration topographique du lieu considéré. En outre, si la fusion des informations GPS avec les données proprioceptives du véhicule (vitesse et orientation des roues) améliore notablement la localisation, le faible niveau de détail de la cartographie du milieu ne permet généralement pas d'estimer, de façon sûre, une trajectoire précise entre les différents points de passage d'un itinéraire. Le processus de navigation oblige donc à construire dynamiquement, ou en post-traitement, un modèle plus détaillé du milieu traversé. Une représentation tridimensionnelle dense est toutefois inutile, puisque le processus de perception pour la conduite, au sein d'un environnement statique, se résume localement à identifier l'espace navigable. Les figures 2.1(a) à 2.1(c) illustrent ainsi la complexité du modèle 3-D des obstacles, dans un contexte urbain, face à la représentation de l'espace libre pour circuler.

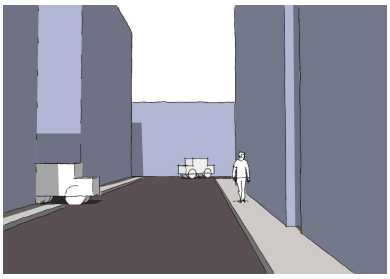
Les véhicules automatisés ont vocation à évoluer en milieu ouvert, et donc vraisemblablement à partager les zones qu'ils desservent avec d'autres modes de locomotion. Ils doivent garantir la sécurité de leurs usagers ainsi que celle des vulnérables[2] à même d'emprunter les voies de circulation. Pour éviter d'entrer en état de "collision inévitable" [2, 3], les modules de planification de trajectoire et de contrôle/commande (sec. 1.2.2) ont besoin, en sus d'un modèle de l'espace libre, des informations relatives aux déplacements des objets mobiles. La représentation minimale de l'environnement doit donc être étendue aux scènes dynamiques avec la description des obstacles en mouvement et de leur état : position et vecteur vitesse (Fig. 2.1(d)).

Le modèle local du monde ainsi défini est indépendant du niveau d'abstraction utilisé pour décrire les objets mobiles et l'espace navigable. Dans le cadre de cette thèse, l'identification de ces deux éléments repose exclusivement sur des critères géométriques, de manière à considérer tout espace suffisamment plan et horizontal comme libre, et tout volume en mouvement comme un obstacle mobile. Il est néanmoins possible d'utiliser d'autres méthodes de classification, afin, par exemple, de limiter la navigation aux routes goudronnées ou encore de différencier les objets

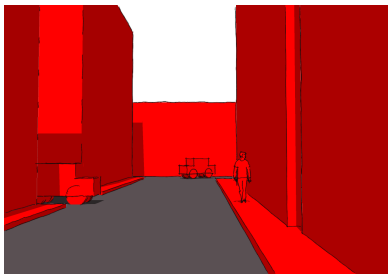
[1] ***G**lobal **P**ositioning **S**ystem.*

[2] Ensemble des êtres vivants susceptibles d'être accidentés.

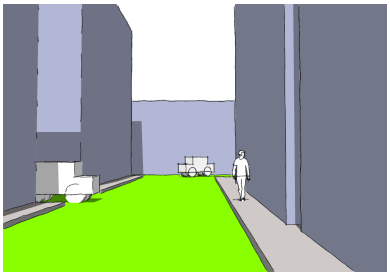
de la scène, selon qu'il s'agisse de vulnérables ou de véhicules.



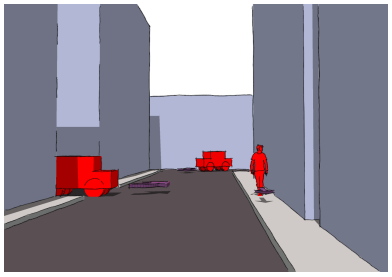
(a) Scène urbaine.



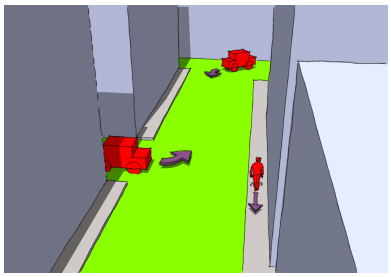
(b) Définition usuelle des obstacles.



(c) Espace navigable.



(d) Obstacles mobiles.



(e) Représentation minimale étendue.

FIG. 2.1 – Modèle de l'environnement.

2.1.2 Les capteurs

Le recueil d'information sur la configuration de l'environnement et l'état du système dépend de l'instrumentation du véhicule et notamment des capteurs qui lui sont intégrés. Cependant, tandis que les données proprioceptives, telles que l'orientation des roues directrices ou la vitesse angulaire des roues motrices, sont mesurées à l'aide de capteurs de contact, la perception de l'environnement en milieu ouvert nécessite de recourir à la télédétection. Ce type de méthode d'acquisition à distance utilise principalement la mesure des rayonnements électromagné-

tiques, émis ou réfléchis dans différentes plages de fréquence, par les objets de la scène. Toutefois, certains instruments emploient, de façon similaire, d'autres formes d'énergie radiative comme les ondes mécaniques sonores. Les dispositifs de télédétection sont classés en deux grandes familles : d'une part, les capteurs passifs, qui recueillent l'énergie émanant naturellement des éléments observés et, d'autre part, les capteurs actifs qui émettent des radiations réfléchies par le milieu puis mesurées au retour.

Les capteurs passifs pour la perception de l'environnement

L'énergie solaire représente la principale source naturelle d'énergie radiative disponible pour les capteurs passifs. Le rayonnement électromagnétique du soleil est absorbé puis diffusé par l'environnement, et l'intégralité du spectre de la lumière visible[3] ainsi réfléchie peut être perçue à l'aide de capteurs photographiques. Quelle que soit la cellule photosensible employée, CCD[4] ou CMOS[5], son fonctionnement repose sur l'effet photoélectrique lié à l'absorption des photons par les photo-diodes dont elle est constituée. Un capteur photographique est toujours associé à un système optique composé d'une ou plusieurs lentilles, l'objectif, pour former une image nette à la surface des photo-diodes. L'ensemble, constitué du capteur et d'un objectif, permet d'acquérir une image de la scène visée, échantillonnée en une matrice de pixels[6], dont les intensités retranscrivent la quantité d'énergie lumineuse reçue par les photo-diodes. Le choix de l'objectif influe sur l'angle de champ, tandis que les propriétés intrinsèques du capteur déterminent la définition et la profondeur de l'image, soit respectivement, ses dimensions en pixels et le nombre de bits codant chacun d'eux. En outre, dans le cas d'une série de prises de vue réalisées pour étudier temporellement la scène, il est nécessaire de considérer la fréquence d'acquisition, dont dépend la discrétisation du mouvement apparent des objets dans le plan image. La notion sous-jacente de *traitement temps-réel* signifie alors que l'intégralité des calculs sur l'image doit être réalisée entre deux acquisitions consécutives. Le contexte applicatif contraint le temps de traitement maximal acceptable et, par conséquent, le choix du capteur photographique en fonction de ses spécifications techniques. Une fréquence d'acquisition comprise entre 15 et 25 hertz, pour une définition comprise entre 320×240 et

[3] Parfois étendu aux infrarouges et aux ultraviolets.
[4] ***C**harge-**C**oupled **D**evice.*
[5] ***C**omplementary **M**etal **O**xide **S**emi-conductor*
[6] ***Pict**ure **El**emen*t. Unité de surface élémentaire d'une image numérique.

640×480 pixels, codés chacun sur 8 bits, constitue à ce jour une configuration type dans le domaine de la vision artificielle pour la robotique mobile.

D'utilisation moins fréquente, le rayonnement infrarouge émis par un corps vivant ou le moteur thermique d'un véhicule, est une autre source d'énergie radiative. Celle-ci peut être détectée à l'aide d'un simple transducteur ou d'une caméra thermique fonctionnant selon le modèle des caméras conventionnelles, à l'aide d'un capteur sensible aux longueurs d'onde supérieures à 745 nanomètres. Offrant l'avantage d'être opérationnel de jour comme de nuit, l'emploi de ce type de rayonnement est cepencdant particulièrement délicat dans certains cas, comme les situations de trop fort rayonnement solaire. Ces capteurs sont caractérisés par leur résolution thermique, indiquant l'écart minimal de température perceptible.

Les capteurs actifs pour la perception de l'environnement

Les capteurs actifs disposent de leur propre source d'énergie pour irradier la scène examinée. Un émetteur dirige le rayonnement de cette énergie vers une cible tandis qu'un récepteur en mesure la réflexion. Sur le principe du radar[7], la connaissance de la vitesse de propagation des ondes émises et du temps écoulé jusqu'à leur retour au capteur, permet de calculer l'éloignement de la cible. La position de celle-ci peut ensuite être estimée en tenant compte de l'orientation de l'émetteur. Parmi les différentes technologies de télédétection employées en robotique mobile, le lidar[8] et, dans une moindre mesure, le radar et le sonar[9], sont les plus usitées. Elles se distinguent entre elles par le domaine spectral dans lequel elles fonctionnent ainsi que par le type de faisceau utilisé.

Le lidar émet une lumière laser impulsionnelle et le point d'impact de chacun de ses tirs est localisé grâce à la réception de son écho. L'exploration systématique de l'environnement est rendue possible grâce à un dispositif de balayage rotatif permettant de scanner jusqu'à 330 degrés. On obtient de cette manière une carte précise, en deux dimensions, des obstacles traversant le plan de tir. A titre d'exemple, le système Ibeo ALASKA (Fig. 2.2) assure une résolution angulaire moyenne de $0,25$ degré pour une fréquence de balayage de $12,5$ hertz, ou $0,5$ degré pour 25 hertz, ainsi qu'une portée effective de près de 60 mètres. Les performances des lidars civils sont essentiellement bornées par certaines normes de sécurité, interdisant l'usage de faisceaux lasers dangereux pour l'oeil humain.

[7] ***RA**dio **D**etection **A**nd **R**anging*, système de radio-repérage.
[8] ***Li**ght **D**etection **A**nd **R**anging.* Parfois LADAR, ***LA**ser **D**etection **A**nd **R**anging.*
[9] ***SO**und **N**avigation **A**nd **R**anging.*

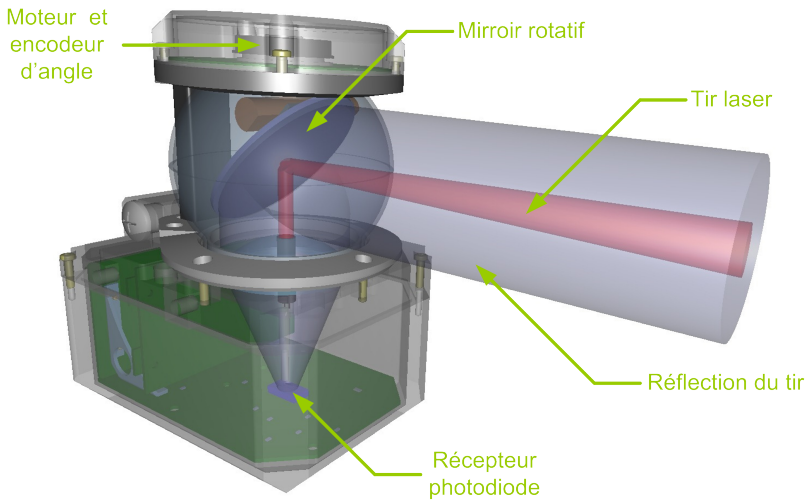


FIG. 2.2 – Schéma d'un lidar (modèle Ibeo ALASKA).

Concernant les capteurs de localisation de type radar ou sonar, les technologies déployées pour leur conception se révèlent être particulièrement coûteuses. Pour limiter leur prix, les dispositifs embarqués, développés dans le cadre des Systèmes de Transport Intelligents (ITS), sont donc conçus pour mesurer uniquement la distance séparant le véhicule du premier obstacle situé dans le cône de rayonnement du capteur considéré. Deux systèmes grand public illustrent leur utilisation : d'une part, le radar de recul, composé en réalité de plusieurs transducteurs à ultrasons (sonars), d'une portée maximale de 3 mètres, répartis de manière à assurer un champ d'action le plus large possible ; d'autre part, le radar anti-collision (section 1.1.3), utile jusqu'à près de 150 mètres grâce aux radio-fréquences des ondes émises, comprises entre 75 et 110 gigahertz.

Quel capteur pour l'automobile ?

A ce jour, scanner laser et caméra conventionnelle demeurent les seules solutions se suffisant à elles-mêmes et capable de suppléer l'automobiliste pour la perception de l'environnement. Par conséquent, ces capteurs sont tous deux couramment utilisés dans le milieu de la recherche sur les véhicules automatisés. A la simplicité d'exploitation des données télémétriques du lidar est opposée la richesse des informations contenues dans une image. Différents critères tendent néanmoins à favoriser l'essor de la caméra au sein des ITS, en vue de l'industrialisation de ces derniers : la résolution des données, la robustesse du capteur ou encore son faible

coût.

La résolution angulaire : Elle permet d'estimer, pour une distance donnée, la taille du plus petit objet détectable. A titre d'illustration, cette taille peut être calculée pour une caméra équipée d'un objectif à focale fixe de 50 millimètres, correspondant approximativement à la vision humaine avec un angle de champ de 47 degrés. Pour une largeur d'image de 640 pixels, on obtient alors une résolution angulaire horizontale de $0,07$ degré, à comparer au $0,25$ degré du lidar. Cette différence se réduit toutefois avec la diminution de la focale utilisée, dès lors qu'il est nécessaire d'augmenter le champ de vision. En outre, tandis que la résolution angulaire verticale de la caméra est identique à sa résolution horizontale, le scanner laser est au mieux doté de différents plans de tir, rarement plus de cinq, séparés chacun de quelques degrés.

La robustesse : La fiabilité du dispositif constitue un second critère de sélection pertinent, notamment lorsqu'il s'agit, à terme, d'intégrer le capteur dans un système embarqué grand public. Une caméra à focale fixe constitue un ensemble solide dénué de liaison mécanique à mobilité non nulle. A l'inverse, le module de balayage d'un lidar comporte nécessairement différentes pièces d'usure. En outre, la télédétection des obstacles, présents devant un véhicule équipé d'un lidar, exige que ce dernier soit fixé au niveau de la calandre, le rendant ainsi vulnérable aux chocs frontaux.

Le coût : Tandis que la part du coût de l'électronique embarquée dans l'automobile ne cesse de croître et correspond aujourd'hui à près de 30% du coût total de fabrication, il n'est pas concevable de commercialiser une voiture équipée d'un lidar dont la valeur doublerait, voire triplerait son tarif de base (en moyenne 21674 euros[10]). A l'opposé, les capteurs photographiques doivent leur succès à leur accessibilité, avec des prix de l'ordre de quelques centaines d'euros.

En définitive, la caméra semble être le capteur adéquat, puisque l'intégration prochaine d'un système de perception de l'environnement dans l'automobile est devenue inéluctable. Malgré son incapacité à fonctionner par trop faible luminosité, elle s'impose grâce à la richesse des informations recueillies, ainsi que par son faible coût qui permet, par ailleurs, la conception de configurations multi-capteurs.

[10]En France sur les modèles neufs, toutes marques confondues. Source : Auto Journal n° 717 (2008).

En outre, la délimitation de l'espace navigable, composante indispensable à la représentation minimale étendue du milieu (section 2.1.1), requiert l'acquisition d'informations suffisamment denses au niveau du sol, que seule la caméra est en mesure de réaliser. Enfin, par analogie avec le système perceptif humain, la caméra semble un capteur suffisant pour percevoir l'environnement dans le cadre des ITS.

2.2 Vision artificielle et perception de l'environnement

La perception est le processus de recueil et de traitement de l'information sensorielle. En vision artificielle, l'information utile, issue des images acquises, peut fortement varier en fonction de la méthodologie employée pour analyser la scène. Toutefois, les pixels qui composent l'image sont toujours regroupés en régions selon des critères pré-établis, dans un procédé de partitionnement appelé "segmentation". L'évolution temporelle des structures, ainsi reconnues et appariées au fil des acquisitions successives, permet la recherche d'invariants projectifs pour construire une représentation plus complexe de l'environnement [64, 65]. Il est également possible de déterminer la profondeur des points de la scène par la mise en correspondance de leurs projections dans différentes prises de vues simultanées. Quelle que soit l'approche employée, l'étude de la géométrie du milieu requiert cependant de connaître les paramètres intrinsèques de la/des caméra(s), à savoir les paramètres de projection de l'espace 3-D dans le(s) plan(s) image(s). Ils sont fixes pour une longueur focale donnée et peuvent donc être obtenus au cours d'une étape préalable de calibration.

La suite de la section résume, dans un premier temps, quelques approches couramment employées en analyse d'image pour la perception de l'environnement et justifie ainsi l'orientation des recherches présentées dans le reste du document. Après cet état de l'art sommaire, une seconde partie explicite les équations nécessaires à la projection des éléments de l'environnement, dans le plan image.

2.2.1 État de l'art

Dans le domaine des transports intelligents, les premiers systèmes de perception de l'environnement par vision artificielle datent de la fin des années quatre-vingt. A cette époque, la puissance de calcul des ordinateurs n'autorisait pas l'étude des modèles tridimensionnels caractérisant la scène observée, tout en respectant les contraintes temporelles nécessaires à la conduite automatisée. Le programme euro-

péen Prométheus, dont notamment les travaux conduits au CMM[11] [12], ainsi que les recherches menées simultanément pour le DARPA[12] [11], sont caractéristiques des approches alors retenues : l'espace navigable est délimité grâce à l'identification des marquages au sol, tandis que les obstacles présents sur la route sont généralement détectés d'après leur symétrie dans le plan rétinien. Bien que ces méthodes se limitaient à l'étude de la géométrie apparente dans l'image, elles nécessitaient souvent de recourir à l'intégration de circuits logiques programmables (FPGA[13]) ou spécifiques (ASIC[14]) [12, 13], afin d'en accélérer les temps d'exécution.

Le bond technologique de ces dernières décennies a cependant permis d'étendre le champs d'action des algorithmes utilisés en robotique mobile temps-réel et l'on discerne aujourd'hui deux grands types d'approche : d'une part les méthodes globales, qui supposent une connaissance *a priori* des éléments de l'environnement, et de l'autre les méthodes locales, qui définissent la scène par l'étude des contraintes liées au modèle projectif du capteur photographique utilisé.

Méthodes globales

Les méthodes globales reposent sur la connaissance, *a priori*, d'un modèle de chacun des éléments recherchés dans la scène. Chaque modèle correspond à la représentation 3-D ou à l'image d'un élément dans le plan focal. On parle alors d'approches *model-based* pour la détection des obstacles [67] et de l'espace navigable [68].

Dans le cas de représentations 3-D, chaque objet virtuel est tout d'abord projeté dans l'image, grâce aux paramètres intrinsèques de la caméra, en fonction d'indices permettant d'initialiser sa pose. Les contours de la projection sont ensuite précisément appariés avec ceux, détectés, de l'image dans un processus d'optimisation, correspondant à la minimisation d'un critère de corrélation. Sur le même principe, C. Cappelle [66] propose ainsi une méthode de géo-localisation par la mise en correspondance d'une cartographie 3-D GIS[15] de l'environnement. Les obstacles sont alors identifiés par comparaison des images virtuelles et réelles de la scène.

Le modèle d'un objet peut également s'apparenter à une base de données, formée d'instances 2-D représentatives de son image réelle. On utilise alors des approches de classification automatique par apprentissage. Une première étape, le

[11] **C**entre de **M**orphologie **M**athématique de l'École des Mines de Paris.
[12] ***D**efense **A**dvanced **R**esearch **P**rojects **A**gency*, département R&D de la Défense des États-Unis.
[13] ***F**ield-**P**rogrammable **G**ate **A**rray.*
[14] ***A**pplication **S**pecific **I**ntegrated **C**ircuit.*
[15] ***G**eographical **I**nformation **S**ystem.*

plus souvent effectuée hors-ligne, détermine, par analyse statistique, une fonction permettant de différencier toute nouvelle entrée parmi les classes de référence de la base de données. Les méthodes de classification non-supervisées utilisent des critères de similitude pour discerner automatiquement les structures d'intérêt photométriques, caractéristiques d'un modèle. A l'inverse, les méthodes supervisées font intervenir un agent extérieur pour labelliser correctement les exemples de la base de données et contrôler l'apprentissage [15]. Parmi elles, les machines à vecteurs de support (SVM[16]) [17] qui rapportent la classification à un problème d'optimisation quadratique ou le *boosting* [16, 18] pour formuler un classifieur fort à partir de plusieurs classifieurs faibles, au moins aussi bons que le hasard.

L'utilisation avec succès des méthodes globales, notamment pour la reconnaissance des véhicules et des piétons, a démontré leur efficacité. Elles demeurent toutefois limitées par l'approximation des modèles employés, qu'il s'agisse de l'imprécision géométrique des représentations synthétiques ou de l'incomplétude de la base d'apprentissage.

Méthodes locales

Les méthodes locales ont pour seule connaissance de la scène une description restreinte, relative au contexte applicatif. La perception monoculaire de l'environnement, à partir d'une image unique, nécessite donc de poser certaines contraintes photométriques et géométriques pour rechercher les régions d'intérêt dans cette dernière. L'espace navigable est ainsi couramment identifié par l'étude du marquage au sol [12, 69], supposant un environnement fortement structuré, ou d'après l'homogénéité colorimétrique des voies de circulation [11, 68, 72]. Les véhicules, souvent assimilés à la notion d'obstacle, sont quant à eux couramment détectés à l'aide d'un critère de symétrie sur les contours de l'image [70, 71].

La stéréoscopie, ou la restitution du relief à partir de deux images planes d'un même sujet, permet de relâcher certaines contraintes imposées par les approches monoculaires. Au sein d'un environnement dynamique, l'acquisition des paires d'images utilisées doit être synchrone. La profondeur relative des points de la scène, communs aux deux images, est retrouvée grâce à la géométrie épipolaire. Elle est obtenue par le calcul de la disparité, à savoir, pour chaque paire de points, la distance en pixels séparant leurs coordonnées dans l'image. Une mesure absolue de la profondeur requiert, en outre, de déterminer la relation liant le repère asso-

[16] ***S**upport **V**ector **M**achine.*

cié à chacune des deux caméras, à l'aide d'une étape préalable de calibration du banc stéréoscopique. La perception du relief par l'estimation dense de la carte des disparités, bien que très largement étudiée [22, 24, 23], reste un compromis entre précision et vitesse d'exécution. C'est pourquoi de nombreux travaux, en accord avec la représentation minimale étendue présentée dans la section 2.1.1, n'ont pas pour objectif de reconstruire exhaustivement la topographie de la scène observée. La transformation de coordonnées IPM[17], par exemple, permet d'identifier les régions de l'image correspond uniquement à l'espace navigable [20, 21]. Il s'agit d'une transformation projective plane, entre le plan du sol et le plan focal, qui supprime les effets de perspective en tout point de l'espace navigable, à la manière d'une vue de dessus. En comparant les transformation IPM appliquée à chaque prise de vue d'une paire stéréoscopique, il est possible de déterminer les points s'élevant au dessus du sol.

L'usage synchronisé de deux caméras peut être remplacé par l'étude des prises de vue successives d'un capteur monoculaire. Dans ce cas, l'estimation en ligne du mouvement de la caméra se substitue à la calibration du banc stéréoscopique. La connaissance du déplacement entre deux acquisitions permet ainsi de rapporter l'analyse géométrique de la scène en monovision aux approches binoculaires mentionnées ci-avant [73]. En simulant la stéréoscopie à l'aide d'un seul capteur photographique en mouvement, la perception de l'environnement se résume à une étude spatio-temporelle de la scène. La compensation de l'*ego-motion*, le mouvement apparent des objets fixes induit par le déplacement de la caméra, permet alors l'identification des obstacles mobiles. Elle nécessite toutefois de considérer le champ de déplacement visuel ainsi estimé comme le mouvement dominant dans l'image, pour ne pas être confondu avec le déplacement propre de ces obstacle. A cette condition, une première famille d'approches propose de circonscrire la recherche des obstacles mobiles aux régions de l'image ayant une dynamique différente d'un modèle approximé du mouvement global. La qualité du résultat dépend alors du modèle de déplacement utilisé ainsi que du type de mouvement observé dans l'image. D'autres approches cherchent à compenser l'*ego-motion* par l'étude d'invariants projectifs [74]. Elles reposent sur le calcul d'un déplacement résiduel qui, associé à la géométrie épipolaire, permet alors de déterminer les éléments mobiles dans la scène.

[17] ***I****nverse* ***P****erspective* ***M****apping.*

Perspective

Le système visuel humain permet de détecter, catégoriser et reconnaître les objets du monde environnant. Son efficacité est telle, qu'à ce jour, aucun dispositif artificiel n'est en mesure de le remplacer. Il sert donc de référence en matière de perception visuelle, notamment dans le domaine de la robotique mobile.

La projection optique de l'environnement sur le plan rétinien permet le plus souvent de reconnaître tout objet précédemment mémorisé. Pour cela, les processus cognitifs liés à la localisation et l'identification des éléments de la scène observée font intervenir des connexions neuronales de type *feedback*. De même, à l'exception de certains modèles de réseaux de neurones entièrement *feedforward*, la majorité des mécanismes d'apprentissage artificielle, dédiées à la perception visuelle, ont une connectivité rétroactive. Ces derniers sont couramment utilisés et donne satisfaction pour la reconnaissance, en ligne, d'obstacles pré-définis, et dans une moindre mesure pour l'identification de l'espace navigable.

Dans le cadre des travaux présentés dans la suite du document, on suppose n'avoir aucune connaissance de l'environnement, de sorte que le discrimination des éléments, jusqu'alors inconnus, repose uniquement sur l'appréciation du relief au travers d'un processus de perception spatio-temporel. Toutefois, les méthodes temps-réel basées sur la géométrie épipolaire se limitent majoritairement à l'analyse statique de la scène, en stéréoscopie, ou posent des hypothèses fortes sur la nature du mouvement apparent et l'absence d'obstacle mobile, dans le cas des approches monoculaires [9]. Aussi est-il nécessaire de comprendre les facteurs responsables de la perception du relief chez l'Homme, afin d'identifier les mécanismes en jeu lors de la conduite automobile. A ce sujet, les travaux menés par Y. Gao [27], permettent de classer ces facteurs en deux catégories, récapitulées dans le tableau 2.1, selon qu'ils soient de type monoculaire ou binoculaire. Hormis les processus de convergence et d'accommodation, difficilement reproductibles à l'aide des capteurs photographiques actuels, ces mécanismes ont fait l'objet de nombreuses études en vision artificielle[7, 5, 6]. La stéréoscopie repose exclusivement sur la disparité binoculaire, définie géométriquement comme le décalage horizontal entre deux images rétiniennes d'un même objet, relatif à la profondeur de celui-ci. Toutefois, l'appréciation de la profondeur se dégrade avec la diminution de la *baseline*, c'est-à-dire l'écart séparant les deux capteurs, et l'augmentation de la distance entre le capteur stéreoscopique et l'objet. Les systèmes de vision artificiel permettent de solutionner ce problème en ajustant l'écartement des caméras selon l'utilisation souhaitée, mais la rigidité du dispositif est d'autant plus difficile

Binoculaire	○ *convergence* ○ disparité binoculaire
Monoculaire	○ *accommodation* ○ parallaxe ○ perspectives géométriques : • perspective linéaire • perspective de texture • perspective de surface • superposition des contours • taille de l'objet connu • ombrage

TAB. 2.1 – Facteurs de la perception du relief chez l'Homme.

à assurer que la *baseline* est importante. Or, l'expérience montre qu'il est possible de conduire sans avoir recours à la stéréoscopie, ce que confirme la directive du Conseil Européen concernant l'aptitude de chacun à prendre le volant : "L'autorité médicale compétente devra (juste) certifier que cette condition de vision monoculaire existe depuis assez longtemps pour que l'intéressé s'y soit adapté"[18].

Les travaux réalisés dans le cadre de cette thèse sont donc consacrés à la conception d'une approche monoculaire pour la construction du modèle minimal étendu. En outre, les récentes avancées technologiques, axées sur la parallélisation du traitement des données à l'aide d'unités de calcul graphique (GPUs[19]), sont prises en considération dans le choix des algorithmes décrits dans les chapitres suivants.

2.2.2 Modèle projectif et mouvement image

L'étude du mouvement image, ou la projection du déplacement réel dans l'image, requiert de formuler les équations de passage entre l'espace euclidien $\mathbb{E}^3$, associé à un repère cartésien dans $\mathbb{R}^3$, et le plan image noté $\mathbb{I}^2$. La suite de cette section introduit donc les notations et formalismes employés pour écrire les transformations dans $\mathbb{R}^3$, avant de poser l'équation du modèle linéaire de la caméra. Enfin, elle expose les difficultés liées au problème sous-dimensionné de la projection inverse de $\mathbb{I}^2$ vers $\mathbb{R}^3$.

[18] Directive du Conseil Européen (1991).
[19] ***G**raphics **P**rocessing **U**nit.*

Transformation 3-D des corps rigides

Dans le cadre de cette étude, les éléments de l'environnement sont tous supposés rigides. Le déplacement de chacun d'eux peut alors être formulé par une unique transformation, g, décrivant le changement de coordonnées de tout ses points, telle que :

$$\begin{aligned} g : \mathbb{R}^3 &\rightarrow \mathbb{R}^3 \\ \mathbf{X} &\rightarrow g(\mathbf{X}) \end{aligned}$$

avec $\mathbf{X} = (X,Y,Z)^T$ les coordonnées cartésiennes d'un point quelconque P de l'objet considéré. Par définition, une transformation rigide[20] préserve les angles et distances, soit le produit scalaire et la norme des vecteurs $\mathbf{u}$ et $\mathbf{v}$, associés aux coordonnées cartésiennes de toute paire de points du solide considéré :

1. (norme) $\| g(\mathbf{u}) \| = \| \mathbf{u} \|, \forall \mathbf{u} \in \mathbb{R}^3$,
2. (produit scalaire) $< g(\mathbf{u}) \mid g(\mathbf{v}) > = g(< \mathbf{u} \mid \mathbf{v} >), \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^3$.

En outre, toute transformation rigide est la combinaison d'une rotation suivie d'une translation. Ainsi, pour tout point M de $\mathbb{E}^3$ avec $\mathbf{X}_1(M) = (X_1,Y_1,Z_1)^T$ et $\mathbf{X}_2(M) = (X_2,Y_2,Z_2)^T$ ses coordonnées cartésiennes, respectivement aux instants t_1 et t_2, l'équation du déplacement de P, dans l'intervalle de temps entre t_1 et t_2, s'écrit donc :

$$\mathbf{X}_2 = \mathbf{R}_{21}\mathbf{X}_1 + \mathbf{t}_{21}\,, \tag{2.1}$$

avec $\mathbf{R}_{21} \in SO(3)$, la matrice de rotation, et $\mathbf{t}_{21} \in \mathbb{R}^3$ le vecteur de translation. On rappelle que $SO(3)$ définit l'espace des matrices du groupe spécial orthogonal de $\mathbb{R}^{3\times 3}$:

$$SO(3) \doteq \left\{ \mathbf{R} \in \mathbb{R}^{\mathbf{3}\times\mathbf{3}} \,|\, \mathbf{R}^T\mathbf{R} = \mathbf{I},\, \det(\mathbf{R}) = 1 \right\}.$$

L'ensemble des transformations rigides $g = (\mathbf{R},\mathbf{t})$ de $\mathbb{R}^3$ est, quant à lui, décrit par l'espace des transformations du groupe spécial euclidien de dimension trois :

$$SE(3) \doteq \left\{ g = (\mathbf{R},\mathbf{t}) \,|\, \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}.$$

Représentation homogène

A l'exception des rotations pures, les transformations de $SE(3)$, telles que formulées dans l'équation (2.1), ont une forme affine. Pour en simplifier l'écriture,

[20] *rigid-body motion.*

notamment lors de l'agrégation des déplacements successifs, il est toutefois possible de linéariser la forme des transformations rigides, en ajoutant une quatrième composante égale à 1, aux coordonnées cartésiennes des points déplacés. Ainsi modifiées, les coordonnées d'un point M de $\mathbb{E}^3$ sont dites "homogènes" et notées $\widetilde{\mathbf{X}}(M) = (X,Y,Z,1)^T \in \mathbb{R}^4$. L'équation (2.1) devient alors :

$$\widetilde{\mathbf{X}}_2 = \widetilde{g}_{21}\widetilde{\mathbf{X}}_1\,, \tag{2.2}$$

avec :

$$\widetilde{g}_{21} = \begin{bmatrix} \mathbf{R}_{21} & \mathbf{t}_{21} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}. \tag{2.3}$$

La transformation rigide inverse est logiquement décrite par :

$$\widetilde{g}_{21}^{-1} = \widetilde{g}_{12} = \begin{bmatrix} \mathbf{R}_{21} & \mathbf{t}_{21} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R}_{21}^T & -\mathbf{R}_{21}^T\mathbf{t}_{21} \\ 0 & 1 \end{bmatrix}.$$

Modèle projectif de la caméra

L'impression des points de $\mathbb{R}^3$ sur le capteur photographique d'une caméra peut être modélisée par la projection centrale illustrée dans la figure 2.3. Dans ce modèle du sténopé linéaire, le processus de formation de l'image repose sur l'intersection des faisceaux lumineux passant par le centre optique de l'objectif avec le plan focal du capteur. Cette représentation suppose néanmoins que le système optique de la caméra, à savoir sa lentille, respecte les conditions de Gauss :

1. Le faisceau doit traverser la lentille au voisinage du centre optique.
2. L'angle d'incidence du rayonnement doit être faible.

Soit $\widetilde{\mathbf{X}}(M) = (X,Y,Z,1)^T$ les coordonnées homogènes d'un point M de l'espace $\mathbb{R}^3$ dans le repère caméra et f la distance focale séparant le centre optique C du plan image. La projection de C dans $\mathbb{I}^2$ définit le centre de l'image, c, de coordonnées $\mathbf{x}(\mathbf{c}) = (u_0, v_0)$. La projection centrale de M sur le plan rétinien, également nommé "plan focal" ou "plan image", s'écrit donc :

$$\begin{aligned} u &= fk_u\frac{X}{Z} + u_0 \\ v &= fk_v\frac{Y}{Z} + v_0 \end{aligned} \tag{2.4}$$

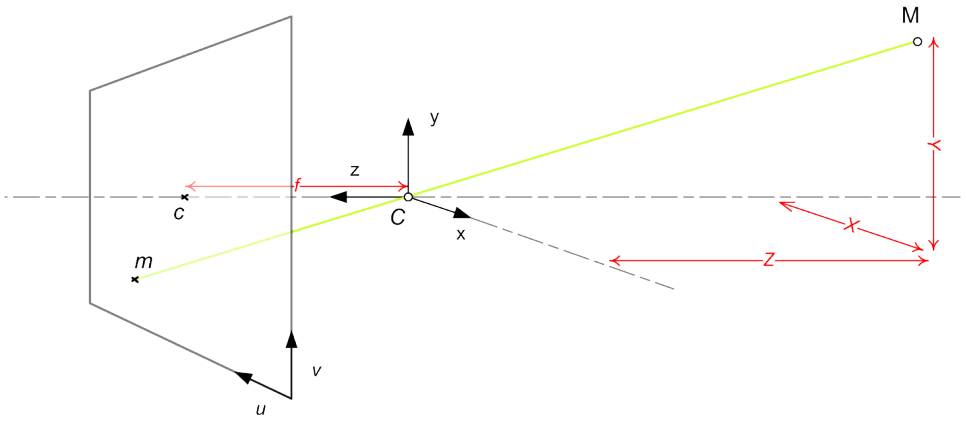


FIG. 2.3 – Modèle du sténopé linéaire.

avec k_v et k_u les facteurs d'échelle horizontal et vertical en pixels par millimètre. En coordonnées homogènes, la transformation de $\mathbb{R}^3$ vers le plan rétinien, d'après le modèle du sténopé linéaire, ou *pin-hole,* se formule également :

$$\widetilde{\mathbf{x}}(m) \propto \mathscr{P}\widetilde{\mathbf{X}}(M)\,, \tag{2.5}$$

avec $\mathscr{P}$ la matrice de projection et $\propto$ désignant l'égalité à un facteur d'échelle près. Le détail de l'équation (2.5) est donné par :

$$Z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\underbrace{\begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathtt{K}}\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Pi_0}}_{\mathscr{P}}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.6}$$

où $\mathtt{K}$ et Π_0 sont respectivement la matrice des paramètres intrinsèques et la matrice de projection canonique. Ainsi, la projection dans le plan focal d'un point M de coordonnées $\mathbf{X}_0(M) = (X_0, Y_0, Z_0)^T$, exprimées dans un repère quelconque $\mathscr{R}_0$, s'écrit :

$$\begin{aligned} Z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} \\ &= \mathtt{K}\Pi_0\, g(\mathbf{R},\mathbf{t})\,\widetilde{\mathbf{X}}_0(M) \end{aligned} \tag{2.7}$$

FIG. 2.4 – Projection inverse depuis une seule image : la position des points de $\mathbb{E}^3$ est estimée à un facteur d'échelle près.

avec $g(\mathbf{R},\mathbf{t})$, la matrice de passage des coordonnées réelles dans le repère caméra.

Projection inverse et mouvement image

L'estimation des coordonnées d'un point M de $\mathbb{E}^3$, à partir de sa projection m_1 dans le plan focal d'une caméra C_1, est un problème sous-dimensionné par définition. Aussi, à partir des équations (2.4), la position recherchée n'est obtenue qu'à un facteur d'échelle près (Fig. 2.4) correspondant à la profondeur du point M :

$$\begin{cases} X = \dfrac{u-u_0}{fk_u}Z \\ \\ Y = \dfrac{v-v_0}{fk_v}Z \end{cases} \tag{2.8}$$

Pour redimensionner correctement le système (2.8), il suffit d'y ajouter les équations de la projection inverse d'un point m_2, image de M dans le plan focal d'une seconde caméra, définie dans le repère lié à C_1. Toutefois, la mise en correspondance des points m_1 et m_2, constitue alors un *a priori* indispensable à la résolution du système. Dans le cas d'acquisitions monoculaires asynchrones, cet appariement équivaut à calculer le mouvement image $\triangle\mathbf{x}(M)$ du point M, à savoir estimer la projection, dans le plan rétinien, de son déplacement réel dans le référentiel asso-

cié au point de vue souhaité :

$$\begin{aligned} \triangle \mathbf{x}(M) &= \mathbf{x}(m_2) - \mathbf{x}(m_1) \\ &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} \mathscr{P}[\widetilde{\mathbf{X}}_2(M) - \widetilde{\mathbf{X}}_1(M)] \end{aligned} \tag{2.9}$$

avec $\widetilde{\mathbf{X}}_1$ et $\widetilde{\mathbf{X}}_2$ les coordonnées homogènes de M, respectivement lors la première puis de la seconde acquisition.

La mise en correspondance de deux acquisitions peut être réalisée grâce à l'intensité lumineuse perçue en chaque point de l'image. Pour cela, on admet, sous certaines conditions, que la luminance des éléments de l'espace euclidien $\mathbb{E}^3$ est invariante dans le temps. En notant $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}_+; \mathbf{x} \rightarrow I(\mathbf{x})$, la représentation fonctionnelle d'une image, qui, à tout point de son domaine de définition Ω, associe une intensité lumineuse, l'équation de conservation de la luminance s'écrit sous forme non linéaire par rapport au mouvement :

$$I_1(\mathbf{x}(m_1)) = I_2(\mathbf{x}(m_2)) = I_2(\mathbf{x}(m_1) + \triangle \mathbf{x}(M)) . \tag{2.10}$$

On appelle alors flot optique, l'estimation du champ de déplacement visuel déterminé d'après cette contrainte. Toutefois, l'équation (2.10) ne permet pas d'approximer le mouvement image de manière unique puisque l'on dispose d'une seule équation pour deux inconnues, les deux composantes du vecteur mouvement. La partie suivante détaille donc, dans un premier chapitre, les différentes approches développées à ce jour pour mesurer le flot optique, dans le cadre des études liées à la robotique mobile.

Deuxième partie

Mouvement image et segmentation

L'analyse d'image effectuée afin d'étudier la perception de l'environnement peut être décomposée entre les approches dites "par apprentissage", les techniques basées sur la recherche d'indices visuels portant sur l'information de profondeur, et enfin les méthodes fondées sur le calcul de contraintes géométriques. Seule cette dernière catégorie ne nécessite pas d'hypothèse forte sur les éléments de l'environnement. Néanmoins, l'étude des caractéristiques 3-D de la scène observée requiert la mise en correspondance des points de l'image entre différentes prises de vue. Ces acquisitions peuvent être synchrones ou successives. Dans le cas des approches monoculaires, l'appariement des points, nécessairement temporel, correspond à l'estimation du mouvement image. Ce dernier est couramment déterminé à partir d'une loi de conservation temporelle de l'intensité des points ou groupes de points dans l'image. De plus, la segmentation de ce mouvement apparent, c'est-à-dire l'identification des régions de mouvement homogène, permet de subdiviser l'image en différentes zones d'intérêt. Une solution algorithmique à ce problème de segmentation, offrant également une mesure de confiance sur les vitesses estimées, consiste à rechercher les surfaces dans l'espace joint 4-D (x, y, v_x, v_y)[21]*. Le chapitre 3 propose tout d'abord une étude des méthodes de calcul du mouvement image, avant de sélectionner la plus adéquate pour le problème étudié. Le chapitre 4 détaille ensuite la question de la segmentation du champ de déplacement visuel.*

[21] (v_x, v_y) le vecteur vitesse au point de coordonnées (x, y).

Chapitre 3

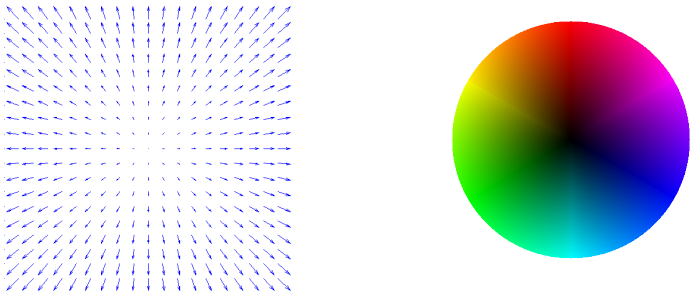
Mouvement image et flot optique

L'appariement des points sur deux images consécutives revient à estimer le mouvement apparent. La sous-détermination du système, issu de la loi de conservation de la luminance et appliqué indépendamment pour chaque point, a conduit au développement de différentes stratégies visant à converger vers une solution unique et stable. Le choix d'une approche dépend du contexte de l'étude, et notamment de la forme du champ de vecteurs associé au mouvement image, ainsi que de l'utilisation *a posteriori* de la solution trouvée. Il est donc important d'évaluer chaque méthode sur une séquence d'images représentative du cadre applicatif.

Dans une première section, le chapitre 3 présente les principaux types d'approches servant à estimer le flot optique. Une seconde section est ensuite réservée à l'évaluation des méthodes satisfaisant les contraintes relatives au processus de perception pour la conduite automatique, à savoir les méthodes temps-réel ou fortement parallélisables et dont le résultat est faiblement lissée au cours du processus d'estimation.

3.1 Définition

L'estimation du mouvement image est un pré-requis indispensable pour de nombreux processus en vision par ordinateur. Sa mesure permet de lever l'ambiguïté relative à la perte d'information occasionnée par la projection de l'environnement dans le plan focal. Elle est donc primordiale en robotique, pour évaluer le déplacement de la caméra ainsi que celui des obstacles mobiles présents dans le champ de vision du capteur. Sans reprendre en détail les explications du chapitre précédent, on rappelle que le mouvement image, ou la projection focale du dépla-



(a) (b)

FIG. 3.1 – Représentations discrète et continue du même flot optique.

cement réel des objets de la scène, est approximé par le mouvement apparent. On parle alors de flot optique, généralement estimé selon l'hypothèse de conservation de la luminance, donnée par la relation (2.10) :

$$I_1(\mathbf{x}_1) = I_2(\mathbf{x}_2) = I_2(\mathbf{x}_1 + \triangle\mathbf{x}) .$$

L'équivalence entre le champ de déplacement visuel et le mouvement image nécessite toutefois la réunion des conditions suivantes :

- un éclairage uniforme de la scène,
- la présence de surfaces Lambertiennes, à savoir de réflectivité isotrope.

Bien qu'en réalité, ce préalable soit rarement établi dans toute la scène, on suppose néanmoins que les conditions requises sont satisfaites localement. Le degré d'approximation du mouvement image dépend alors de la vraisemblance de ces hypothèses.

Le flot optique est classiquement représenté de manière éparse, à l'aide des vecteurs de déplacement d'un échantillon de points de l'image. Cette description intuitive du déplacement apparent est appropriée pour étudier la norme et l'orientation du mouvement, mais ne permet pas de discerner certaines caractéristiques du champ de vecteurs vitesse, dont notamment la continuité spatiale. Il est donc fréquent d'employer le code couleur illustré sur la figure. 3.1(b), qui sera utilisé dans la suite du document. L'orientation et la norme du déplacement sont alors respectivement données par les composantes H (la teinte) et V (la valeur) de l'espace colorimétrique HSV.

3.1.1 Modèle continu du flot optique

En supposant infinitésimale la durée dt séparant deux acquisitions, il est possible d'écrire un modèle continu de l'équation du flot optique. A cet effet, on note $\omega \in \mathbb{R}^2$ le vecteur vitesse associé à un point, de manière à poser l'égalité suivante : $\triangle \mathbf{x} = \omega dt$. L'équation discrète du flot optique (2.10) devient donc :

$$I(\mathbf{x}(t),t) = I(\mathbf{x}(t) + \omega dt, t + dt) .$$

Le développement de la série de Taylor correspondante, autour de $\mathbf{x}(t)$ et jusqu'au terme d'ordre 1, donne :

$$I(\mathbf{x}(t) + \omega dt, t + dt) = I(\mathbf{x}(t),t) + \nabla I(\mathbf{x}(t),t)^T \omega + I_t(\mathbf{x}(t),t) + \mathscr{O}(1) ,$$

avec I_x, I_y, I_t, les dérivées partielles de l'image et

$$\nabla I(\mathbf{x}(t),t) = (I_x(\mathbf{x}(t),t), I_y(\mathbf{x}(t),t))^T$$

le gradient spatial. En négligeant les termes d'ordre supérieur, l'écriture compacte de l'équation de conservation de la luminance s'écrit alors :

$$\nabla I^T \omega + I_t = 0 . \tag{3.1}$$

Il est également possible de déduire directement cette équation de l'hypothèse de conservation de l'intensité lumineuse au cours du temps, avec $dI/dt = 0$. En effet :

$$\begin{aligned} \frac{dI}{dt} &= \frac{dI(x(t),y(t),t)}{dt} \\ &= I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t \\ &= \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} dx/dt \\ dy/dt \end{bmatrix} + I_t \\ &= \nabla I^T \omega + I_t \end{aligned}$$

Toutefois, l'estimation du flot optique, d'après l'équation (3.1), est un problème mal posé. L'équation de conservation de l'intensité, ou équation de contrainte du mouvement apparent, est insuffisante pour déterminer localement les deux composantes spatiales de $\omega \in \mathbb{R}^2$: seule la composante normale ω_∇ du mouvement, dans la direction du vecteur gradient, peut réellement être évaluée. Ainsi, en définissant

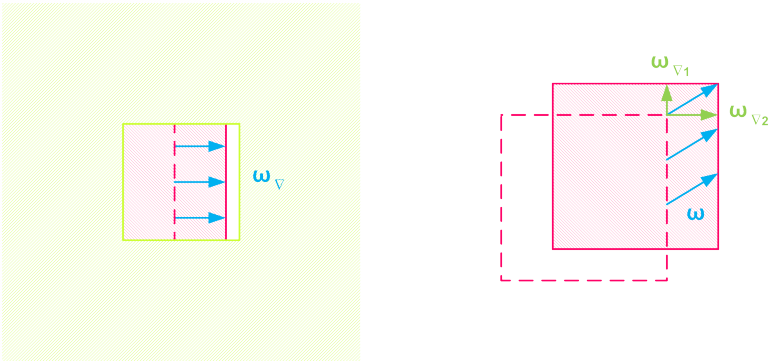


FIG. 3.2 – Problème d'ouverture lié à la perception locale du mouvement apparent.

le vecteur directeur du gradient $\mathbf{n}_\nabla = \nabla I / \|\nabla I\|$, on obtient :

$$\omega_\nabla = (\omega \cdot \mathbf{n}_\nabla) \cdot \mathbf{n}_\nabla = \frac{-I^T \nabla I}{\|\nabla I\|^2} \tag{3.2}$$

Il s'agit du problème d'ouverture, illustré par la Fig. 3.2 : lorsque le gradient spatial de la région d'intérêt Ω_{ROI}, sur laquelle est intégrée la contrainte du flot optique (3.1), est orienté uniformément dans une direction différente de celle du déplacement réel, le mouvement image n'est plus correctement apprécié. Il est nécessaire d'élargir l'ouverture jusqu'à lever l'ambiguïté.

3.1.2 Techniques d'optimisation

Le calcul du flot optique peut être étudié comme un problème d'optimisation, visant à estimer le mouvement image par la minimisation de la différence des intensités lumineuses entre une image, transformée selon le déplacement calculé, et l'acquisition suivante de la séquence considérée. Les causes d'erreur sont principalement :

- le bruit dans l'image, en rapport avec la sensibilité du capteur photographique,
- le problème d'ouverture précédemment énoncé,
- les occlusions et les objets mobiles,
- l'aliasing temporel induit par une fréquence d'acquisition trop faible,
- l'aliasing spatial induit par l'échantillonnage numérique de l'image.

Cependant, quelle que soit la méthode utilisée pour résoudre l'équation du flot optique, il existe différentes stratégies pour pallier à certaines des difficultés énoncées et converger vers une solution unique et stable.

Contraintes multiples Pour trouver la composante tangentielle du mouvement apparent, il est nécessaire de réduire l'espace des solutions par l'introduction de contraintes supplémentaires. L'utilisation d'un espace colorimétrique multidimensionnel consiste à écrire l'équation (3.1) séparément pour chaque composante, afin de contraindre suffisamment la solution lorsque ces dernières ne sont pas corrélées entre elles. Une approche semblable consiste à imposer la conservation temporelle de plusieurs grandeurs, telles que le contraste ou les dérivées partielles de l'image, pour construire un système multi-contraintes à l'aide d'une équation de type (3.1) pour chaque grandeur. Plus couramment, on impose une contrainte de continuité spatiale sur le champ de vecteurs vitesse, afin que chaque point et son voisinage, soient régis par un modèle de déplacement connu (affine, homographique, constant, etc.).

Approche hiérarchique La majorité des méthodes de mesure du mouvement apparent ont une déclinaison multi-échelles. Une approche pyramidale (Fig. 3.3) permet en effet d'augmenter l'amplitude des déplacements mesurables tout en limitant le coût de calcul de la mise en correspondance entre deux acquisitions. Le principe repose sur la réduction des images plutôt que sur l'augmentation de l'ouverture. Les images sont sous-échantillonnées sur plusieurs niveaux à l'aide d'un masque gaussien. Le flot optique est ensuite estimé au niveau le plus élevé, avant d'être interpolé bi-linéairement au niveau inférieur, afin d'y initialiser le calcul du mouvement résiduel, en centrant la fenêtre de travail Ω_{ROI} à l'extrémité du vecteur de déplacement résultant. Ce processus est répété jusqu'au niveau 0, pour lequel les dimensions de l'image sont celles d'origine. La taille relative de l'ouverture, par rapport aux dimensions de l'image et en fonction du niveau considéré, permet d'établir une première estimation du déplacement, au sommet de la pyramide, puis de la préciser en descendant jusqu'à la base. Le gain offert par une approche pyramidale, sur la norme maximale des déplacements calculés, est donné par le facteur suivant :

$$g(l) = (2^l - 1)$$

où l désigne le nombre de niveaux.

Raffinement itératif et temporel Le raffinement itératif consiste à réaliser successivement plusieurs estimations du flot optique, chacune initialisée, de la même façon que pour l'approche pyramidale, à l'aide du champ de déplacement précédemment calculé. Lorsque le système est correctement posé, il converge vers une

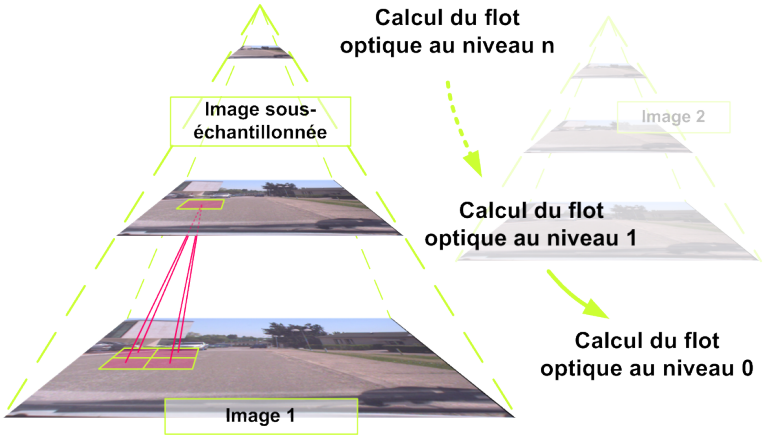


FIG. 3.3 – Implémentation pyramidale de l'estimation du flot optique.

solution stable et le mouvement résiduel décroît alors au fil des itérations. L'algorithme de calcul du flot optique est ainsi répété jusqu'à une condition d'arrêt, fonction du nombre d'itérations réalisées ou de la norme du mouvement résiduel estimé à l'itération courante.

Lorsque l'initialisation du processus de calcul du flot optique, à l'instant t, est obtenue par l'estimation du mouvement image faite à $t-1$, on parle alors de raffinement temporel. Contrairement au raffinement itératif qui peut être couplé efficacement à une approche pyramidale, en étant répété à chaque étage de la pyramide, le raffinement temporel n'est d'aucune utilité pour initialiser l'estimation du flot optique au sommet de cette dernière : le sous-échantillonnage du champ de déplacement, calculé à $t-1$, ne permet pas à celui-ci d'influencer significativement la convergence du processus d'estimation du mouvement image.

Mouvements images multiples L'occultation d'un élément de la scène, dans le plan focal, par l'image d'un second élément, induit l'impossibilité de calculer le déplacement apparent des points occultés entre deux prises de vue, à l'aide des méthodes classiques d'estimation du flot optique. Le phénomène d'occlusion est à l'origine de la plupart des discontinuités du mouvement apparent, observées dans l'étude de séquences d'images réelles. Afin d'en prévenir les effets au regard des contraintes de continuité spatiale nécessaires à l'estimation du flot optique, certaines approches relaxent ces contraintes le long des lignes de plus fort gradient[1] [31, 28, 29, 30] tandis que d'autres proposent un modèle de mouvement

[1] *Oriented smoothness constraint.*

image à multiples distributions de vitesse[2] [32]. Plus simplement [33] emploie un filtre de Kalman pour intégrer temporellement les vitesses estimées.

Mesure de confiance Malgré les nombreux algorithmes développés à ce jour, l'estimation du flot optique reste une estimation partiellement erronée du mouvement image. L'ensemble des problèmes précédemment énoncés éclaire ainsi sur la nécessité d'évaluer le champ de déplacement visuel calculé à l'aide d'une mesure d'erreur. A cet effet, la grande majorité des techniques d'appréciation du flot optique sont fondées, non pas sur le déplacement estimé, mais sur les propriétés intrinsèques de l'image, comme la magnitude du gradient spatial, le déterminant de la matrice hessienne[3] ou encore la matrice de covariance de l'intensité. Il s'agit alors d'estimer localement si l'image permet de contraindre correctement l'estimation du flot optique et notamment sa composante normale (Eq. (3.2)) en chaque point. Toutefois, d'autres critères peuvent être utilisés, dont certains, directement sur les valeurs du champ de vecteurs vitesse calculé. Le chapitre 4 présente à ce sujet une solution robuste, basée sur la continuité spatiale du mouvement image.

3.2 Estimation du mouvement apparent

La mesure du flot optique est un sujet largement traité en vision par ordinateur. La suite du chapitre énumère donc, de façon non exhaustive, les différents types d'approche employés en robotique mobile. Il s'agit de méthodes généralistes, n'utilisant aucune contrainte ni modèle de déplacement comme connaissance *a priori* sur la nature du mouvement image. Pour de plus amples détails concernant ces algorithmes, le lecteur est invité à se référer aux articles suivants : [35, 37, 55].

3.2.1 Approches par corrélation

L'appariement par bloc, ou *block matching*, repose sur une mesure de similitude par région entre deux images. Le mouvement apparent $\omega = (\omega_x, \omega_y)^T$, en chaque point $\mathbf{x} = (x, y)^T$, est obtenu pour la corrélation maximale entre une fenêtre, ou bloc, Ω_{ROI}, centrée sur $\mathbf{x}$ dans l'image originale I_1, et la fenêtre correspondante dans I_2, translatée selon ω. On emploie généralement l'un des critères de similitude suivants :

[2] *Mixed velocity distributions.*
[3] Dérivée partielle seconde de $I(\mathbf{x}, t)$.

– Le minimum de la somme des valeurs absolues (SAD[4]),

$$\sum_{\Omega_{ROI}} |I_1(x,y) - I_2(x+\omega_x, y+\omega_y)| .$$

– Le minimum de la somme des carrés (SSD[5]),

$$\sum_{\Omega_{ROI}} (I_1(x,y) - I_2(x+\omega_x, y+\omega_y))^2 .$$

– Le maximum de la covariance croisée normalisée (NCC[6]),

$$\frac{1}{|\Omega_{ROI}| - 1} \sum_{\Omega_{ROI}} \frac{\left(I_1(x,y) - \bar{I}_1\right)\left(I_2(x+\omega_x, y+\omega_y) - \bar{I}_2\right)}{\sigma_1 \sigma_2} ,$$

où $\bar{I}_i$ et σ_i désignent respectivement la moyenne et l'écart type des intensités dans la fenêtre associée à l'image i.

En pratique, la recherche du déplacement apparent pour un point considéré est limitée spatialement afin de restreindre le nombre des calculs effectués. Différentes stratégies d'exploration de la fenêtre de recherche ont été développées [45, 46, 47, 48], la plus simple restant une recherche exhaustive sur un voisinage borné des coordonnées de la fenêtre de référence. Toutefois, puisque les composantes du déplacement séparant le centre des blocs corrélés sont mesurées en valeurs entières sur la matrice de l'image, le *block matching* ne fournit qu'une estimation pixelique du mouvement apparent. La précision peut être améliorée en sur-échantillonnant l'image, par interpolation bi-linéaire ou par l'emploi d'un filtre plus sophistiqué tel que le *six-tap filter,* utilisé dans le standard $H.264$ pour la compression vidéo [49]. Augmenter la précision d'un facteur deux multiplie alors le nombre de calculs par quatre.

3.2.2 Approches fréquentielles

L'espace des fréquences est couramment utilisé en vision artificielle pour réduire la complexité algorithmique de certains processus, tel que la convolution d'un filtre. Le passage au domaine fréquentiel s'effectue alors par la transformée de Fourier discrète sur l'image. Les méthodes fréquentielles d'estimation du flot optique nécessitent de caractériser le mouvement dans l'espace des fréquences.

[4] ***S**um of **A**bsolute **D**ifferences.*
[5] ***S**um of **S**quared **D**ifferences.*
[6] ***N**ormalized **C**ross **C**orrelation.*

On note $I(x,y,t)$ l'image acquise à l'instant t et $\hat{I}(f_x,f_y,f_t)$ sa transformée de Fourier, avec f_x, f_y et f_t les fréquence spatio-temporelles. La translation discrète $\omega(t) = (\omega_x,\omega_y)^T$ appliquée à toute l'image, sous la contrainte de conservation de la luminance au cours du temps, permet d'écrire :

$$I(x,y,t) = I_0(x+\omega_x t, y+\omega_y t)\,. \tag{3.3}$$

avec $I_0 = I(x,y,0)$. La transformée de Fourier de (3.3) donne :

$$\hat{I}(f_x,f_y,f_t) = \hat{I}_0(f_x,f_y)\cdot\delta(ft-\omega_x f_x-\omega_y f_y)\,,$$

où δ représente la distribution de Dirac. Cette expression montre comment le spectre d'énergie pour une image 2-D en translation a des valeurs nulles partout à l'exception d'un plan passant par l'origine avec pour équation :

$$\omega_x f_x+\omega_y f_y-f_t=0\,. \tag{3.4}$$

Ce plan d'énergie non nulle, ou plan de vitesse, traduit l'équation du flot optique, dans l'espace des fréquences.

Méthode par filtrage La localisation du plan de vitesse peut être réalisée au moyen des filtres spatio-temporels orientés de Gabor. Il s'agit de filtres 3-D passe-bandes, notés $\mathscr{G}(x,y,t)$, formés par le produit d'une gaussienne d'écart-type $(\sigma_x,\sigma_y,\sigma_t)$ et d'une fonction trigonométrique :

$$\mathscr{G}(x,y,t) = \frac{1}{(2\pi)^3\sigma_x\sigma_y\sigma_t}e^{\left(-\frac{x^2}{2\sigma_x}-\frac{y^2}{2\sigma_y}-\frac{t^2}{2\sigma_t}\right)}\cdot\cos\left(2\pi\left(f_{x0}x+f_{y0}y+f_{t0}t\right)\right)\,,$$

où (f_{x0},f_{y0},f_{t0}) représente la fréquence centrale du filtre $\mathscr{G}$. Heeger [39] propose une approche hiérarchique, dans laquelle chaque niveau de la pyramide est convolué à l'aide d'une famille de douze filtres de Gabor W_i, d'orientations et de fréquences temporelles différentes. Il estime ainsi le déplacement au sens des moindres carrés sur la réponse énergétique $\hat{I}_i = \hat{I}\star W_i$ de ces filtres, de manière à satisfaire l'équation fréquentielle du flot optique (3.4). Le principal inconvénient de ce type de méthode est inhérent à la perte d'information entraînée par le filtrage de l'image, nécessaire à la caractérisation du déplacement, et qui peut peut être trop important.

Il existe par ailleurs différentes techniques d'estimation du flot optique fondées sur la transformée en ondelettes [43, 42, 44], dont notamment l'approche de C. Bernard [44], qui projette l'équation du flot optique sur une base d'ondelettes orthogonales. Le système est ensuite résolu en posant l'hypothèse de séparation d'échelle, c'est-à-dire en considérant le déplacement constant sur les quatre ondelettes de la base. Ces méthodes sont équivalentes aux décompositions sur des familles de filtres et induisent une paramétrisation difficile, due au choix *a priori* des ondelettes.

Méthode par corrélation de phase Une mise en oeuvre efficace des fonctions de corrélation est également possible dans le domaine de Fourier. C.D. Kuglin et D.C. Hines [41] suggèrent ainsi d'utiliser le domaine fréquentiel pour calculer la covariance croisée normalisée, en multipliant la première image par le conjugué complexe de la seconde acquisition :

$$\mathscr{R}_{t,t+1} = \frac{\hat{I}_{t+1}\hat{I}_t^*}{\left|\hat{I}_{t+1}\hat{I}_t^*\right|} . \tag{3.5}$$

La surface de corrélation de phase est ensuite déterminée par la transformée inverse du *"cross power spectrum"* (3.5) :

$$r_{t,t+1} = \mathfrak{F}^{-1}\left(\mathscr{R}_{t,t+1}\right) .$$

On obtient alors l'estimation du mouvement image par :

$$(\omega_x, \omega_y) = \underset{(x,y)}{argmax}\left\{r_{t,t+1}\right\} .$$

Le calcul du flot optique se limite donc à la mesure pixelique du déplacement dominant pour chaque région étudiée dans l'image. En outre, s'il existe des variantes sous-pixeliques de cette technique [40], la norme du déplacement est néanmoins toujours bornée par les dimensions $M \times N$ de la région considérée, de sorte que $\omega_{max} = \left({}^{M}/_{2}; {}^{N}/_{2}\right)^T$.

3.2.3 Approches variationnelles

Méthodes Globales L'estimation du mouvement apparent peut également correspondre à la résolution d'un problème d'optimisation, reposant sur la minimisation d'une fonctionnelle, basée sur l'équation du flot optique (3.1) et associée à

une contrainte assurant, si possible, une solution unique et stable. Les méthodes dites globales introduisent pour cela un terme de régularisation sur le champ des vitesses estimées et réalisent la minimisation sur l'ensemble de l'image. Cette régularisation porte le plus souvent sur le gradient du déplacement apparent, avec une fonctionnelle qui s'écrit alors :

$$\int_{\Omega} \left(\nabla I^T \omega + I_t\right)^2 + \alpha \left(\phi \left(\|\nabla \omega_x\|\right) + \phi \left(\|\nabla \omega_y\|\right)\right) d\mathbf{x}.$$

Le premier terme reprend la contrainte de conservation de l'intensité, tandis que la fonction $\phi(\cdot)$ contrôle le lissage du champ de vitesses et est déterminée afin de préserver au mieux les discontinuités du flot optique. Les équations d'Euler-Lagrange résultantes font apparaître un terme de divergence, formé par la relation linéaire de deux composantes, l'une dans la direction du gradient et l'autre, dans la direction orthogonale [57]. Lorsque $\phi\left(\|\nabla \omega_x\|\right) = \|\nabla \omega_x\|^2$, la pondération de ces deux composantes est identique et le lissage isotrope [56]. On retrouve alors la solution initialement proposée par B. K. P. Horn et B. G. Schunck [34] :

$$\int_{\Omega} \left(\nabla I^T \omega + I_t\right)^2 + \alpha \left(\|\nabla \omega_x\|^2 + \|\nabla \omega_y\|^2\right) d\mathbf{x}. \tag{3.6}$$

Le réglage du coefficient de pondération α nécessite de prendre en considération le type de séquence vidéo étudié, tandis que le nombre d'itérations nécessaires à la minimisation de (3.6) dépend de l'étendue des régions de gradient nul.

Méthodes Locales A l'inverse, les méthodes locales résolvent l'équation du flot optique sur une zone de l'image pour laquelle la nature du mouvement apparent est connue. Lucas et Kanade [36] supposent la vitesse ω uniforme sur un voisinage Ω_{ROI}, de sorte qu'il soit possible de résoudre, au sens des moindres carrés, l'équation (3.1) sur Ω_{ROI}. L'estimation du déplacement apparent est alors calculée par le minimisation de la fonctionnelle suivante :

$$\sum_{\Omega_{ROI}} W^2(\mathbf{x}) \left(\nabla I\, \omega + I_t\right)^2, \tag{3.7}$$

où $W(\mathbf{x})$ représente une fonction de pondération, couramment modélisée par une distribution gaussienne.

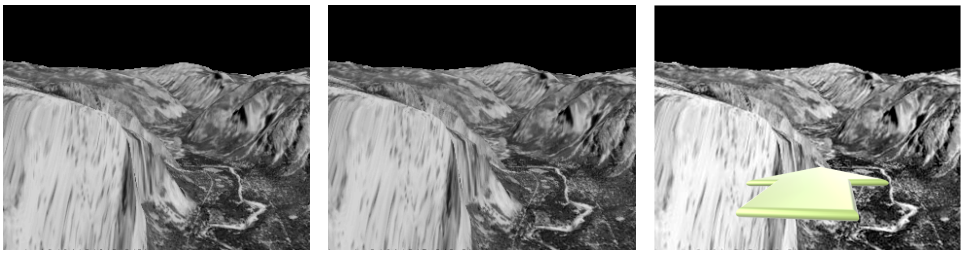
Comme pour les méthodes globales et l'algorithme de Horn et Schunck, il existe plusieurs variantes du modèle local proposé par Lucas et Kanade, exhausti-

vement détaillées dans l'état de l'art concernant le calcul du flot optique présenté dans [37]. Les méthodes variationnelles locales permettent finalement de mieux apprécier les discontinuités du flot optique dans l'image, mais contraignent l'utilisateur à régler la taille du voisinage utile Ω_{ROI}, en fonction de l'amplitude maximale des déplacements à mesurer. L'approche hiérarchique détaillée à la section 3.1.2 réduit notablement cette difficulté.

3.3 Étude comparative

La suite du chapitre présente une étude comparative des approches précédemment décrites, afin de sélectionner la méthode la mieux adaptée au problème de perception de l'environnement pour les systèmes de transport intelligents. Aussi convient-il de définir un certain nombre de critères pour les départager, et l'on rappelle, à cette occasion, que l'estimation du déplacement des points de l'image, entre deux acquisitions, a pour but l'identification de l'espace navigable et des obstacles mobiles (*i.e.* la représentation minimale étendue de l'environnement). On ne dispose en outre, d'aucun *a priori* sur la nature du déplacement des obstacles, ni sur leur géométrie ou celle des contours de l'espace libre dans le plan focal. La segmentation de l'image doit donc tenir compte de contraintes dynamiques, plutôt que photométriques, et reposer sur une mesure dense (en chaque point de l'image) du flot optique. Il est nécessaire d'obtenir une estimation sous-pixelique du champ de déplacement visuel afin d'identifier correctement les discontinuités du mouvement image induites par le déplacement relatif des régions connexes. En conséquence, les méthodes par corrélation dans le domaine de Fourier ne sont pas retenues. On écarte également les méthodes par filtrage spatio-temporel ainsi que les approches par transformée en ondelettes, qui induisent un lissage trop important des images ainsi qu'une paramétrisation délicate. On privilégie les algorithmes temps-réel ou largement parallélisables de manière à garantir à court terme l'opérabilité temps-réel du processus final de perception de l'environnement. Les approches retenues pour cette étude comparative se limitent donc aux méthodes variationnelles globales et locales, ainsi qu'au *block-matching*.

La comparaison des différentes mesures du flot optique est réalisée sur une séquence vidéo synthétique en niveaux de gris, pour laquelle le mouvement image exact est connu. Cette séquence témoin (Fig. 3.4(a)), couramment utilisée pour évaluer l'estimation du mouvement apparent, permet de mesurer précisément l'influence des paramètres propres à chaque méthode sur le déplacement calculé. Pour



(a) Séquence de synthèse Yosemite. *Le mouvement de la caméra, entre les deux images de gauche, est indiqué par une flèche verte.*

(b) Séquences de référence Route. *Le mouvement de la caméra, entre les deux images de gauche, est indiqué par une flèche verte tandis que celui des obstacles mobiles est noté par une flèche rouge.*

FIG. 3.4 – Séquences vidéos témoins.

cela, on établit généralement la moyenne sur l'image de l'erreur angulaire (AAE[7]) ainsi que la moyenne de la norme de l'erreur, respectivement formulées par :

$$AAE = \frac{1}{N \cdot M} \sum_N \sum_M \arccos\left(\frac{u_c u_r + v_c v_r + 1}{\sqrt{\left(u_c^2 + v_c^2 + 1\right)\left(u_r^2 + v_r^2 + 1\right)}} \right)$$

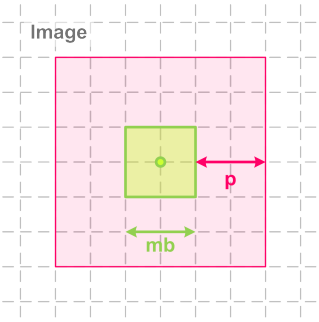
et :

$$Norm = \frac{1}{N \cdot M} \sum_N \sum_M \sqrt{\left(u_c - u_r\right)^2 + \left(v_c - v_r\right)^2}.$$

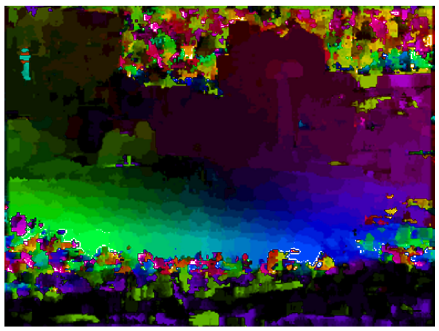
$N \times M$ désigne la taille des images tandis que $(u_r, v_r)^T$ et $(u_c, v_c)^T$ représentent les déplacements image réel et estimé.

Cette séquence ne simule toutefois pas l'observation du milieu depuis un véhicule de transport routier. Il est donc nécessaire d'opérer une comparaison complémentaire à l'aide d'images réelles, contextuellement plus représentatives. On emploie pour cela la vidéo d'une caméra embarquée dans un véhicule roulant sur une voie dénuée de marquage au sol, tandis qu'un véhicule progresse en sens inverse, parallèlement à l'axe optique du capteur (Fig. 3.4(b)).

[7] ***A**verage **A**ngular **E**rror.*



(a) Ouverture et fenêtre de recherche.

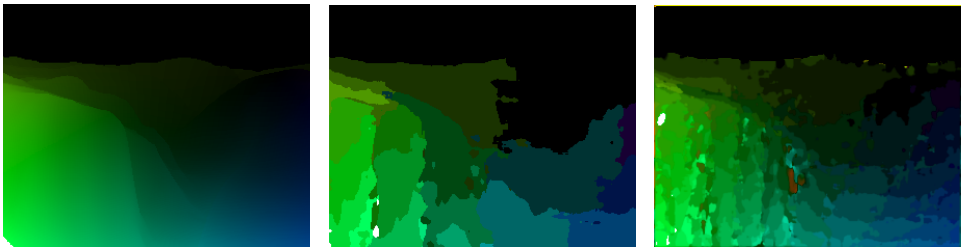


(b) Estimation dense.

FIG. 3.5 – Estimation dense du flot optique par *block-matching*, pour la séquence *Route* (mb=16, p=8).

3.3.1 *Block matching*

Quel que soit l'algorithme de *block matching* considéré, la vitesse est supposée constante par bloc et l'on effectue une recherche de plus grande similitude sur la seconde image, au voisinage du bloc de référence. La différence de position entre le patch initial et le meilleur candidat donne alors la mesure du mouvement apparent. La taille *mb* du patch et l'étendue *p* de la zone de recherche sont les deux paramètres immuables à toutes les techniques de *block matching* (Fig. 3.5(a)). Le premier équivaut à l'ouverture tandis que le second permet d'augmenter la taille maximale des déplacements détectés. Une mesure dense du flot optique peut être réalisée en considérant pour chaque point de l'image, le bloc centré sur la position du point. Les meilleurs résultats sont obtenus par une mesure exhaustive de la corrélation au sein de la fenêtre de recherche. La figure 3.5(b) illustre cette mesure pour la séquence *Route* convertie en niveaux de gris : l'estimation du déplacement apparent est globalement cohérent dans toute l'image, à l'exception des régions de couleur uniforme ou des zones occultées dans la seconde acquisition. L'estimation sous-pixelique du flot optique peut-être obtenue en déplaçant le bloc, pour lequel on suppose le mouvement image uniforme, de manière sous-pixelique également. La mesure de similitude est alors réalisée par interpolation de la luminance des points de l'image courante aux coordonnées entières du bloc déplacé. Cependant, on multiplie ainsi par quatre le nombre de valeurs de similitude à calculer, pour une fenêtre de recherche et un patch de tailles identiques. Selon le même principe, il est nécessaire d'effectuer seize fois plus de calculs pour atteindre une précision d'un $1/4$ de pixel, et ainsi de suite. La figure. 3.6 permet de comparer vi-



(a) Mouvement image réel. (b) BM pixelique. (c) BM sous-pixelique.

FIG. 3.6 – Estimation dense du flot optique par *block-matching*, pour la séquence témoin *Yosemite* (mb=16, p=8).

suellement différents résultats, obtenus par corrélation de bloc, avec le mouvement image réel de la séquence *Yosemite*. Malgré une estimation au $1/2$ pixel près, on observe très nettement les discontinuités induites par la quantification de l'estimation du déplacement apparent. Le tableau 3.2 récapitule l'erreur moyenne calculée pour chaque estimation du flot optique, et sert de référence pour l'évaluation des approches suivantes.

	Erreur angulaire (deg)	Erreur sur la norme ($pixel(s)$)
BM (mb=16, p=8), pixelique	4.5001	0.3889
BM (mb=16, p=8), sur-échantillonné une fois	3.1126	0.3410

TAB. 3.2 – Mesure de l'erreur des algorithmes de *block-matching* pour la séquence *Yosemite*.

En 1995, C.Q. Davis, Z.Z. Karu et D.M. Freeman établissent le parallèle entre les approches variationnelles et les méthodes de *block-matching*, dont notamment les algorithmes étudiés cette section. Ils démontrent ainsi que tout algorithme de *block-matching* 2-D, sous-pixelique par sur-échantillonnage bi-linéaire, utilisant la somme des différences au carré comme mesure de similitude, peut être remplacé de manière équivalente par un algorithme variationnel employant des dérivées du premier ordre, et réciproquement [50].

3.3.2 Méthodes variationnelles

Horn et Shunck

La méthode d'estimation du flot optique de Horn et Schunck combine l'équation de conservation de la luminance avec un terme de régularisation servant à lisser le champ de déplacement visuel mesuré. La fonctionnelle (3.6) ainsi formulée peut être minimisée par la résolution des équations d'Euler-Lagrange équivalentes :

$$\begin{cases} I_x\left(I_x\omega_x + I_y\omega_y + I_t\right) - \alpha\triangle\omega_x &= 0 \\ I_y\left(I_x\omega_x + I_y\omega_y + I_t\right) - \alpha\triangle\omega_y &= 0 \end{cases}$$

où $\triangle(\cdot)$ représente le laplacien, à savoir l'opérateur différentiel égal à la somme des dérivées partielles du second ordre. Le résultat est alors obtenu itérativement par la méthode de Jacobi :

$$\begin{cases} \omega_{x,n+1} &= \overline{\omega}_{x,n} - \dfrac{I_x\left(I_x\overline{\omega}_x + I_y\overline{\omega}_y + I_t\right)}{\alpha^2 + I_x^2 + I_y^2} \\ \omega_{y,n+1} &= \overline{\omega}_{y,n} - \dfrac{I_y\left(I_x\overline{\omega}_x + I_y\overline{\omega}_y + I_t\right)}{\alpha^2 + I_x^2 + I_y^2} \end{cases} \tag{3.8}$$

avec $\omega_{n+1} = \left(\omega_{x,n+1}, \omega_{y,n+1}\right)^T$ le déplacement estimé à l'itération $n+1$ ($\omega_0 = \overrightarrow{0}$) et $\overline{\omega}$ le déplacement moyen au voisinage du point considéré. Le nombre d'itérations nécessaire à la résolution des équations (3.8) est dépendant de la dimension des surfaces dont la luminance est uniforme. L'estimation du flot optique en ces points est en effet propagée itérativement depuis les régions voisines. Une approche hiérarchique permet naturellement de réduire le nombre d'itérations, en diminuant la taille des régions d'intensité homogène de l'image dans les étages supérieurs de la pyramide. Le modèle pyramidal présente également l'avantage de rendre la méthode moins sensible au bruit dans l'image, grâce à l'interpolation bi-linéaire entre chaque niveau. La figure 3.7 montre ainsi les champs de vitesse obtenus pour 100 itérations, avec un (Fig. 3.7(b)) puis trois niveaux (Fig. 3.7(c)) de sous-échantillonnage, sur la séquence *Route* convertie en niveaux de gris. Le paramètre de lissage α, utilisé pour réaliser ces estimations, a été déterminé par l'étude empirique de son influence sur les erreurs angulaires et sur la norme, à partir de la séquence de référence *Yosemite* (Fig. 3.8). Si l'on constate un flot globalement cohérent, avec notamment l'observation d'un champ de vitesse radial sur la figure. 3.7, le caractère isotrope de la régularisation donne un résultat très lissé,

(a) Séquence originale. (b) Sur 1 niveau. (c) Sur 5 niveaux.

FIG. 3.7 – Estimation du flot optique selon l'approche pyramidale de l'algorithme proposé initialement par Horn et Schunck, sur la séquence *Route*.

avec des discontinuités peu marquées aux frontières des éléments mobiles de la scène.

Lucas et Kanade

Lucas et Kanade considèrent un mouvement image constant sur le voisinage Ω_{ROI} de chaque point de l'image. Le déplacement d'un point est ainsi donné par la minimisation de la fonctionnelle (3.7) suivante :

$$\sum_{\Omega_{ROI}} \left(\nabla I\, \omega + I_t\right)^2 ,$$

D'où :

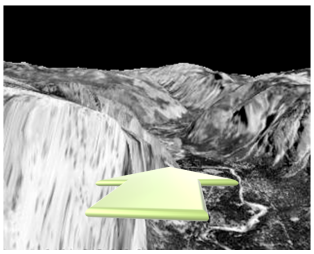
$$\underbrace{\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}}_{\omega} = - \underbrace{\begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \\ I_{tn} \end{bmatrix}}_{\mathbf{b}} . \tag{3.9}$$

Le système ainsi formé peut être résolu au sens des moindres carrés :

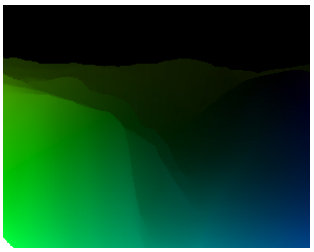
$$A^T A \omega = A^T \left(-\mathbf{b}\right) .$$

Ainsi :

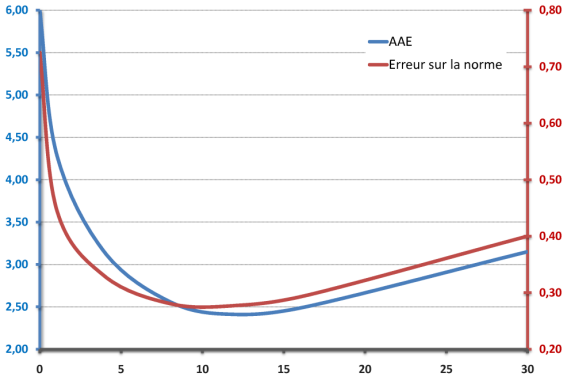
$$\omega = \left(A^T A\right)^{-1} A^T \left(-\mathbf{b}\right) ,$$



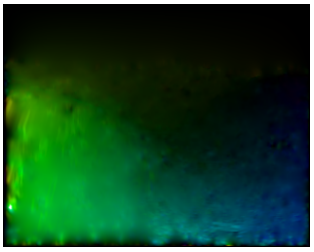
(a) Séquence originale.



(b) Déplacement réel.



(c) Erreur angulaire et sur la norme.



(d) Déplacement estimé.

FIG. 3.8 – Estimation du flot optique selon Horn et Schunck sur la séquence *Yosemite* et calcul de l'erreur en fonction du coefficient de régularisation α.

ou encore :

$$\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = \begin{bmatrix} \sum\limits_{\Omega_{ROI}} I_x^2 & \sum\limits_{\Omega_{ROI}} I_x I_y \\ \sum\limits_{\Omega_{ROI}} I_x I_y & \sum\limits_{\Omega_{ROI}} I_y^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum\limits_{\Omega_{ROI}} I_x I_t \\ -\sum\limits_{\Omega_{ROI}} I_y I_t \end{bmatrix}. \tag{3.10}$$

Lorsque le gradient de l'image au voisinage du point considéré est nul, la matrice $A^T A$ est mal conditionnée (déterminant nul) et donc non-inversible. Néanmoins, une approche pyramidale peut résoudre le problème si les régions de luminance uniforme n'ont pas une surface trop importante. L'ouverture Ω_{ROI}, constante sur tous les niveaux de la pyramide, permet en effet de considérer, dans le système (3.9), les zones de gradient non nul, voisines du point étudié dans l'image sous-échantillonnée courante. Le fonctionnement de l'approche hiérarchique de l'algorithme de Lucas et Kanade à raffinement itératif est illustré sur trois niveaux dans la Fig 3.9. Le choix de l'ouverture, la taille du voisinage Ω_{ROI} utilisé dans l'équation (3.10), est réalisé grâce à l'analyse des erreurs angulaires ou de norme,

pour la séquence de référence *Yosemite*. Le graphique des résultats obtenus pour un sous-échantillonnage sur trois niveaux, est donné Fig. 3.10. Les deux mesures d'erreur apparaissent nettement corrélées, avec un minimum obtenu pour un voisinage de 11×11 pixels.

L'estimation du flot optique sur la séquence d'images réelles, *Route*, pour cette même ouverture, est illustrée Fig. 3.11. L'obstacle mobile représenté par la voiture roulant en sens inverse est clairement identifiable grâce aux discontinuités marquées à son contour. En outre, le champ de déplacement radial induit par le mouvement propre de la caméra est également nettement visible au niveau du sol. En revanche, l'approche se révèle sensible au bruit. Ainsi, l'estimation du déplacement apparent est erronée dans les zones statiques, telles que la zone des essuie-glaces du véhicule embarquant le capteur ou lorsque le gradient est quasiment nul, par exemple au niveau du ciel.

Les capteurs photographiques modernes sont, pour la plupart, capables de différencier certaines longueurs d'onde du spectre lumineux pour fournir des images en couleur de la scène. Cette information colorimétrique est couramment représentée dans l'espace (R,G,B), des intensités lumineuses liées aux composantes rouge, verte et bleue. Elle peut être intégrée de différentes manières au calcul du flot optique, le plus simple étant de retrouver l'internsité lumineuse par combinaison linéaire des tois canaux R, G et B. On lisse ainsi le bruit de mesure inhérent à chacune des trois cellules du capteur photographique. De cette façon, l'erreur résiduelle des mesures aberrantes est diminuée, rendant plus robuste la résolution du système (3.9) par les moindres carrés.

Une seconde solution, largement référencée dans la littérature ([51, 52]), interprète les différentes composantes colorimétriques comme autant d'images en niveaux de gris. L'équation du flot optique sur une paire d'images s'écrit alors en chaque point :

$$\begin{cases} I_x^R \omega_x + I_y^R \omega_y + I_t^R &= 0 \\ I_x^G \omega_x + I_y^G \omega_y + I_t^G &= 0 \\ I_x^B \omega_x + I_y^B \omega_y + I_t^B &= 0 \end{cases} . \tag{3.11}$$

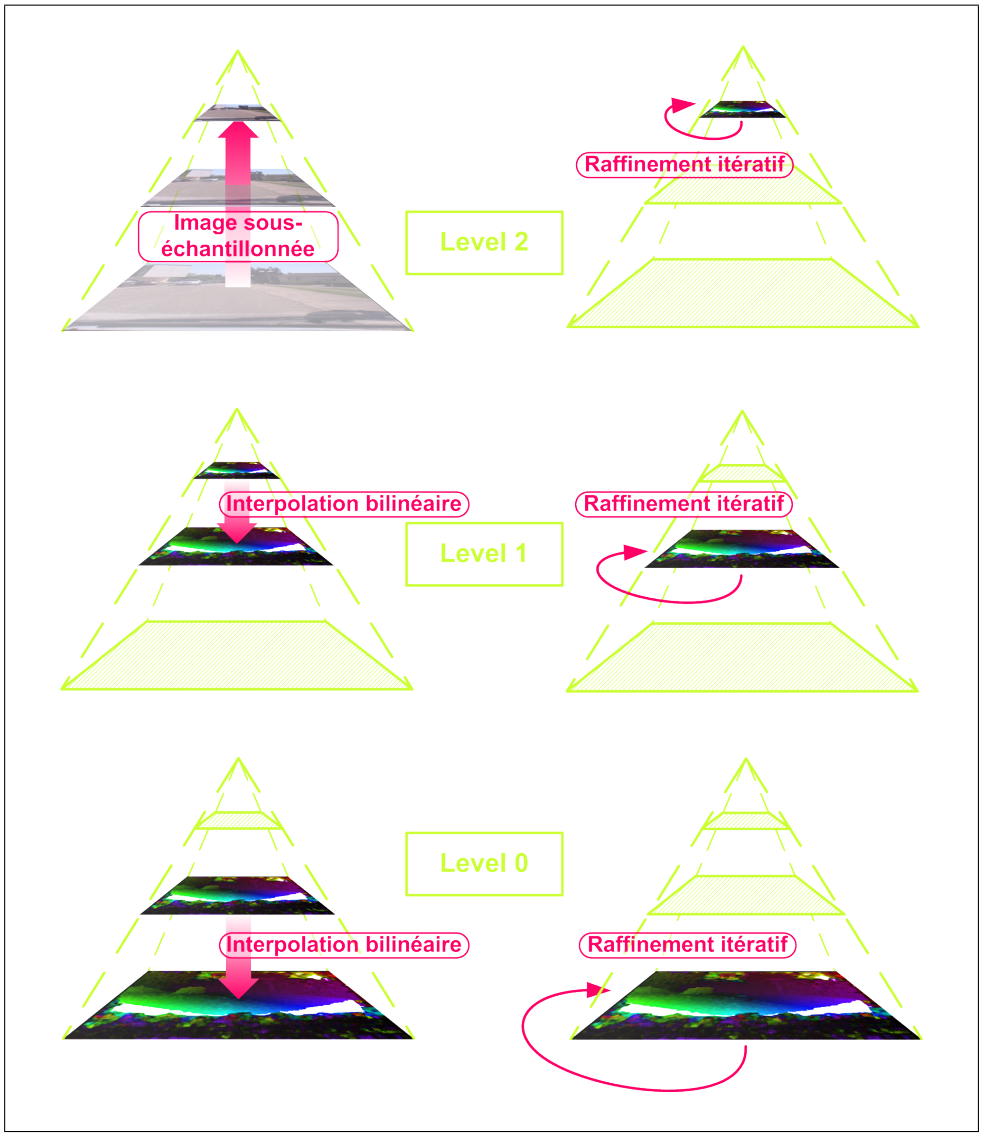
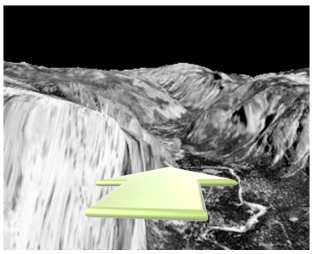
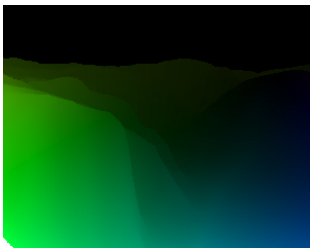


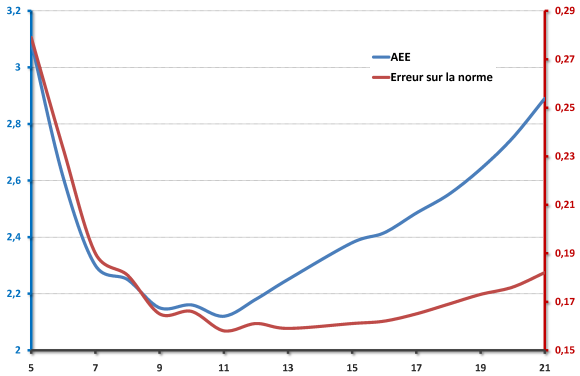
FIG. 3.9 – Schéma sur 3 niveaux de l'approche hiérarchique pour l'algorithme de Lucas et Kanade à raffinement itératif.



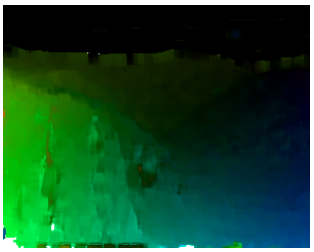
(a) Séquence originale.



(b) Déplacement réel.



(c) Erreur angulaire et sur la norme.



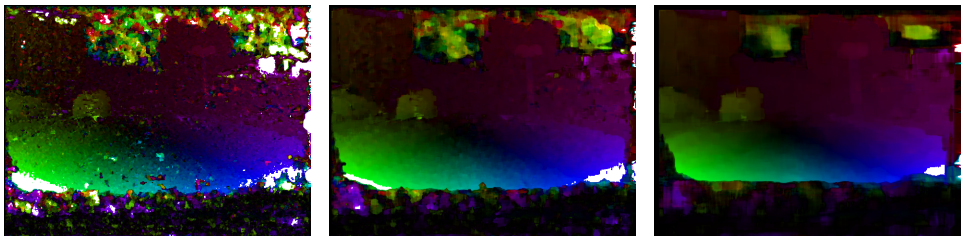
(d) $\Omega_{ROI} = 10 \times 10$

FIG. 3.10 – Estimation du flot optique selon Lucas et Kanade, sur la séquence *Yosemite*, et calcul de l'erreur en fonction de la taille de l'ouverture.

Le système (3.9) devient donc :

$$\underbrace{\begin{bmatrix} I_{x1}^R & I_{y1}^R \\ I_{x1}^G & I_{y1}^G \\ I_{x1}^B & I_{y1}^B \\ I_{x2}^R & I_{y2}^R \\ \vdots & \vdots \\ I_{xn}^B & I_{yn}^B \end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}}_{\omega} = - \underbrace{\begin{bmatrix} I_{t1}^R \\ I_{t1}^G \\ I_{t1}^B \\ I_{t2}^R \\ \vdots \\ I_{tn}^B \end{bmatrix}}_{\mathbf{b}},$$

et se résout au sens des moindres carrés pour estimer le déplacement apparent de manière analogue à la méthode originale, avec $\omega = \left(A^T A\right)^{-1} A^T (-\mathbf{b})$. L'algorithme de Lucas et Kanade peut ainsi être appliqué aux images couleurs, non seulement dans l'espace (R,G,B), mais également dans les espaces colorimétriques



(a) $\Omega_{ROI} = 4 \times 4$ (b) $\Omega_{ROI} = 10 \times 10$ (c) $\Omega_{ROI} = 20 \times 20$

FIG. 3.11 – Estimation du flot optique d'après la méthode pyramidale de Lucas et Kanade, sur la séquence *Route*.

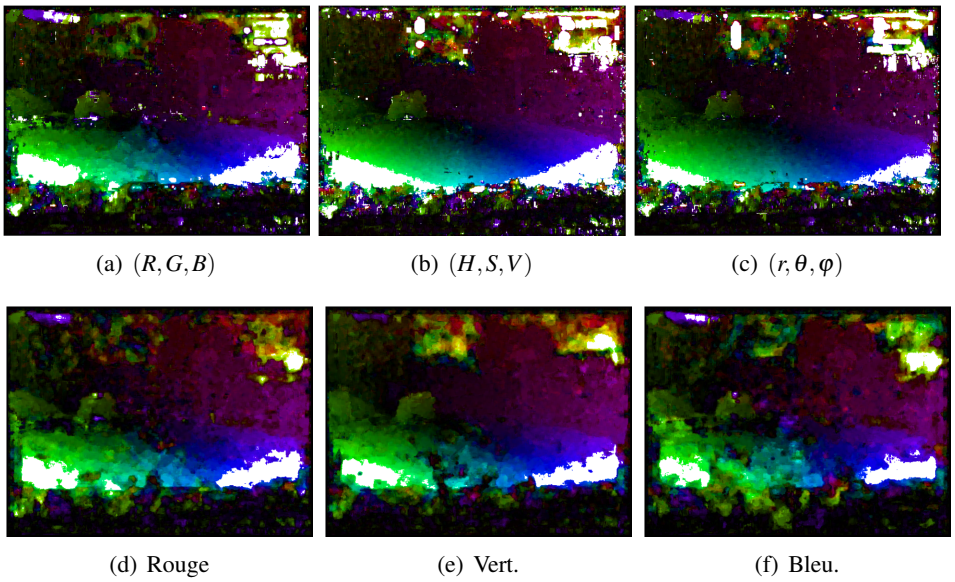
(r, θ, φ) :

$$\begin{aligned} r &= \sqrt{R^2+G^2+B^2} \\ \theta &= \arctan\left(\frac{G}{R}\right) \\ \varphi &= \arcsin\left(\frac{\sqrt{R^2+G^2}}{\sqrt{R^2+G^2+B^2}}\right) \end{aligned}$$

ou (H, S, V) :

$$\begin{aligned} H &= \begin{cases} 0 & \text{, si max} = \text{min} \\ \left(60 \cdot \dfrac{G-B}{\max(R,G,B) - \min(R,G,B)}\right) \bmod 360 & \text{, si max} = R \\ 60 \cdot \dfrac{B-R}{\max(R,G,B) - \min(R,G,B)} + 120 & \text{, si max} = G \\ 60 \cdot \dfrac{R-G}{\max(R,G,B) - \min(R,G,B)} + 240 & \text{, si max} = B \end{cases} \\ S &= \begin{cases} 0 & \text{, si max} = 0 \\ \dfrac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)} & \text{, sinon} \end{cases} \\ V &= \max(R,G,B) \end{aligned}$$

J. Barron et R. Klette [51] présentent une étude quantitative de l'estimation couleur du flot optique, dans laquelle sont comparées, entre autre, les deux approches précédemment décrites. La conclusion de cette étude préconise l'utilisation séparée des composantes de la couleur (seconde méthode). Pourtant, les résultats de la figure. 3.12, obtenus dans différents espaces de couleur, sur la séquence d'images réelles *Route*, paraissent moins cohérents que celui estimé sur les images préalablement converties en niveaux de gris (Fig. 3.11(b)). Une analyse plus précise de l'étude menée par J. Barron et R. Klette montre que les séquences synthé-



(a) (R,G,B) (b) (H,S,V) (c) (r,θ,φ)

(d) Rouge (e) Vert. (f) Bleu.

FIG. 3.12 – Estimation du flot optique, selon Lucas et Kanade, dans différents espaces colorimétriques.

tiques utilisées sont construites d'après une seule image, sans introduction de bruit. Elles ne simulent donc aucune erreur de mesure, et la résolution du système (3.11), au sens des moindres carrés, n'est biaisée par aucun point aberrant.

Afin de ne pas avoir à gérer la sensibilité au bruit de mesure, des méthodes d'estimation du flot optique par les moindres carrés, la suite du document utilise la somme pondérée des composantes R, G et B, qui donne l'intensité lumineuse en niveaux de gris et lisse le bruit propre à chaque canal de couleur (Fig. 3.12(d), (e) et (f)). Selon ce principe, on convertit donc les acquisitions d'une caméra couleur en niveaux de gris plutôt que d'utiliser nativement un capteur noir et blanc.

3.3.3 Conclusions

La solution apportée par l'algorithme pyramidal à raffinement itératif, de Lucas et Kanade, semble la mieux adaptée à l'étude du champ de déplacement apparent dans le cadre des ITS. En l'absence de modèle *a priori* du mouvement image, elle offre, sans lissage excessif, une précision suffisante pour percevoir les discontinuités du flot optique, induites par le déplacement relatif des régions connexes à segmenter. Couramment employée dans le secteur de la recherche, notamment par le laboratoire d'intelligence artificiel de Stanford, l'implémentation proposée dans la librairie de traitement d'image OpenCV, fait figure de référence. Toutefois, son

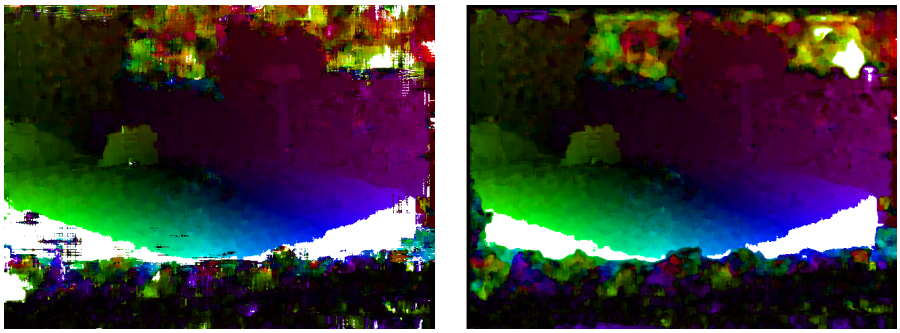


FIG. 3.13 – Flot optique estimé à l'aide de la librairie OpenCV (à gauche) et notre version (à droite).

interpolation "au plus proche voisin", lors du sur-échantillonnage des vecteurs vitesse entre deux niveaux de la pyramide, laisse apparaître quelques artefacts, sous la forme d'une trame visible par endroit dans l'image de la Fig. 3.13. On décide donc d'implémenter l'algorithme de Lucas et Kanade en utilisant une interpolation bi-linéaire pour cette étape. Par ailleurs, le temps d'exécution pour mesurer le flot optique sur trois niveaux, à partir d'une image de 640×480 pixels, est d'environ trois secondes sur un processeur mono-core de 3 GHz. Aussie, le chapitre 7 propose une solution matérielle et logicielle pour assurer l'estimation du mouvement image en temps réel.

Chapitre 4

Tensor Voting et segmentation du flot optique

Le mouvement relatif des objets dans l'image traduit leur profondeur relative dans la scène ainsi que leur mouvement propre. La segmentation du flot optique consiste à reconnaître ces objets par le partitionnement de l'image en ensembles de points connexes et de déplacement homogène, soit continu à la discrétisation près du domaine spatial de définition de l'image. Pour cela, on identifie les discontinuités du mouvement apparent qui correspondent à la frontière de ces régions. Ces régions pourront elles-mêmes être agrégées en fonction de la nature de leur déplacement, dans un second processus de segmentation décrit aux chapitres 5 et 6.

Le champ de déplacement visuel apporte une information multidimensionnelle, qui associe les composantes des vecteurs vitesse $\mathbf{v} = (v_x, v_y)^T$ aux coordonnées des points de l'image $\mathbf{x} = (x, y)^T$. Une dimension temporelle peut également être ajoutée, de manière à prendre en compte la date d'acquisition des images par le capteur photographique. Dans ce contexte, la théorie des tenseurs [58] offre des outils mathématiques adaptés pour donner une métrique cohérente à cette information et en simplifier le traitement, et l'analyse.

La suite du chapitre définit, dans une première partie, la notion de tenseur symétrique, non-négatif, et présente une première méthode de segmentation du flot optique basée sur le vote de tenseurs 3-D. Elle développe ensuite le formalisme du *Tensor Voting*, élaboré par G. Medioni, avec une seconde méthode de partitionnement basée sur les propriétés topographiques de l'espace 4-D (x, y, v_x, v_y). Dans les deux cas, le résultat est une image des discontinuités du mouvement apparent. La segmentation de cette image, pour définir les régions dont le mouvement est

homogène, est détaillée dans la dernière section.

4.1 Les tenseurs symétriques définis positifs du second ordre

4.1.1 Définitions et propriétés

Un tenseur T est une application multilinéaire définit sur un K-espace vectoriel V, à valeur dans $\mathtt{K}$:

$$T : \underbrace{V* \times \cdots \times V*}_{r} \times \underbrace{V \times \cdots \times V}_{s} \longrightarrow \mathtt{K}$$

avec $V*$ l'espace dual associé aux formes linéaires de V. Le tenseur T d'ordre $r+s$ ainsi défini, est dit r-fois contravariant et s-fois covariant.

Par définition, lorsque V désigne un espace Euclidien, à savoir de dimension finie n et muni d'une loie de composition interne appelée produit scalaire, tout vecteur $\mathbf{x}$ de cet espace peut être exprimé comme la combinaison linéaire des vecteurs d'une base $\mathscr{B} = \{\mathbf{e}_i\}_{1 \leqslant i \leqslant n}$ de V tel que :

$$\mathbf{x} = x^i \mathbf{e}_i$$

où chaque scalaire x^i correspond à une composante contravariante de $\mathbf{x}$. La base duale $\mathscr{B}* = \left\{\mathbf{e}^j\right\}_{1 \leqslant j \leqslant n}$, définie de la manière suivante :

$$\mathbf{e}_i \mathbf{e}^j = \delta_i^{\ j} \qquad avec\ \delta_i^{\ j} = \begin{cases} 0 & , i \neq j \\ 1 & , i = j \end{cases}$$

permet d'écrire le vecteur $\mathbf{x}$ dans l'espace dual, à l'aide des ses composantes covariantes x_j :

$$\mathbf{x} = x^i \mathbf{e}_i = x_j \mathbf{e}^j$$

Ainsi :

$$\begin{array}{rcccccc} \mathbf{x}\mathbf{e}^k & = & x^i \mathbf{e}_i \mathbf{e}^k & = & x^i \delta_i^k & = & x^k \\ \mathbf{x}\mathbf{e}_k & = & x_j \mathbf{e}^j \mathbf{e}_k & = & x_j \delta_k^j & = & x_k \end{array}$$

Les coordonnées covariantes sont donc lues, dans le repère lié à $\mathscr{B}$, comme la projection orthogonale de $\mathbf{x}$ sur les axe du repère. Aussi, lorsque la base considérée est orthonormée, elle est confondue avec son dual. La suite du chapitre traite exclusivement de ce cas.

Un tenseur d'ordre k est représentée dans un espace vectoriel de dimension n, par n^k composantes associées à une base de cet espace. En conséquence, un tenseur de rang zéro correspond à un scalaire, tandis que les composantes des tenseurs d'ordre supérieur sont déterminées par l'action de ces derniers sur les vecteurs de la base orhonormée $\mathscr{B} = \{\mathbf{e}_i\}_{1\leqslant i\leqslant n}$ considérée. Soit f un tenseur du premier ordre exprimé en coordonnées cartésiennes dans $\mathscr{B}$ par $a_i = f(e_i), \forall i \in [1;n]$. La fonction linéaire f appliquée à tout vecteur $\mathbf{x}$, s'écrit alors :

$$f(\mathbf{x}) = f\left(\sum_n x_i \mathbf{e}_i\right) = \sum_n x_i f(\mathbf{e}_i) = \sum_n a_i x_i = \left(\sum_n a_i \mathbf{e}_i\right)\mathbf{x} = \mathbf{a}\cdot\mathbf{x}.$$

On représente ainsi les tenseurs du premier ordre par un covecteur $\mathbf{a}$, dont l'écriture matricielle correspond à un vecteur ligne, ramenant alors le produit scalaire à un simple produit matricielle. De manière analogue, la matrice $n \times n$ associée à la forme bilinéaire désignée par un tenseur du second ordre T, s'écrit dans le repère lié à $\mathscr{B}$:

$$\mathbf{T} = \begin{bmatrix} T_{11} & \cdots & T_{1n} \\ \vdots & & \vdots \\ T_{n1} & \cdots & T_{nn} \end{bmatrix},$$

avec $T_{ij} = T(\mathbf{e}_i, \mathbf{e}_j) = \mathbf{e}_i^T \mathbf{T}\mathbf{e}_j$.

L'algèbre tensorielle, ou algèbre multilinéaire, est définie sur la base des outils vectoriels. Aussi, les propriétés des tenseurs du second ordre sont directement issues de la théorie des matrices. Quel que soit l'espace n-dimensionnel considéré, il existe donc pour certains tenseurs du second ordre T, des vecteurs particuliers $\mathbf{v}_i$ tels que :

$$\mathbf{T}\mathbf{v}_i = \lambda \mathbf{v}_i, \tag{4.1}$$

avec λ_i la *valeur propre* associée au *vecteur propre* $\mathbf{v}_i$, de forme normalisée notée $\hat{\mathbf{v}}_i = \dfrac{\mathbf{v}_i}{\|\mathbf{v}_i\|}$. L'équation vectorielle (4.1) peut également s'écrire sous la forme d'un système d'équations sur les composantes du vecteur propre :

$$T_{ij} v_j = \lambda v_i,$$

pour lequel il existe des solutions non-triviales v_i, si et seulement si :

$$\det(\mathbf{T} - \lambda \mathbf{I}) = 0. \tag{4.2}$$

Dans le cas d'un tenseur symétrique, avec $T_{ij} = T_{ji}$, le polynôme (4.2) possèdent

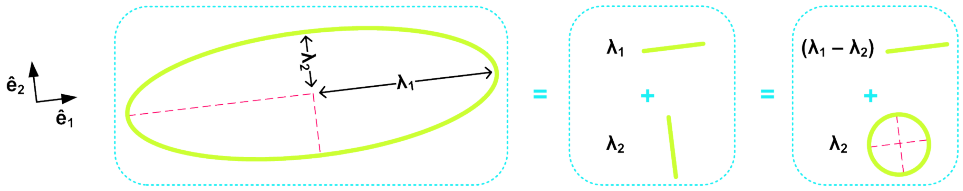


FIG. 4.1 – Représentation graphique d'un tenseur symétrique générique du second ordre, défini, non-négatif, et de sa décomposition en tenseurs élémentaires.

n racines réelles $\{\lambda_i | \lambda_i \geqslant \lambda_{i+1}\}_{1\leqslant i\leqslant n}$ associées leurs vecteurs propres unitaires, $\{\hat{\mathbf{e}}_i\}_{1\leqslant i\leqslant n}$, orthogonaux entre eux :

$$\mathbf{T}\hat{\mathbf{e}}_i = \lambda\hat{\mathbf{e}}_i \qquad 1 \leqslant i \leqslant n\,,$$

d'où la décomposition matricielle suivante :

$$\begin{aligned}\mathbf{T} &= \mathbf{T}\mathtt{I} = \mathbf{T}\left(\hat{\mathbf{e}}_i\hat{\mathbf{e}}_i^T\right) = \sum_n \lambda_i\hat{\mathbf{e}}_i\hat{\mathbf{e}}_i^T \\ &= (\lambda_1-\lambda_2)\,\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + (\lambda_2-\lambda_3)\left(\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T\right) + \cdots + \lambda_i\left(\sum_n \hat{\mathbf{e}}_i\hat{\mathbf{e}}_i^T\right)\end{aligned} \tag{4.3}$$

La suite du chapitre se restreint à l'étude des tenseurs du second ordre, *symétriques, définis, non-négatifs*, à savoir des tenseurs symétriques de valeurs propres supérieures ou égales à zéro.

En notant $\mathbf{x}_0$ l'origine de la base propre $\mathscr{B} = \{\hat{\mathbf{e}}_i\}_{1\leqslant i\leqslant n}$ d'un tenseur T, la représentation géométrique de ce dernier est donnée par l'image de la boule unité fermée $b(\mathbf{x}_0, 1) = \{\mathbf{x} \in V / \|\mathbf{x}-\mathbf{x}_0\| \leq 1\}$ par l'endomorphisme :

$$T(\mathbf{x}) = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}\mathbf{x} \tag{4.4}$$

exprimé dans $\mathscr{B}$. Dans le cas d'un espace bi-dimensionnel, l'équation (4.4) décrit une ellipse de demi-diamètres λ_1 et λ_2, orientés dans la direction des vecteurs propres $\hat{\mathbf{e}}_1$ et $\hat{\mathbf{e}}_2$, tandis que la surface quadrique définit pour $n = 3$ est alors une ellipsoïde. La figure 4.1 illustre la décomposition selon l'équation (4.3) d'un tenseur symétrique défini positif, générique, en *tenseurs élémentaires* pondérés. Ces formes dégénérées sont appelées *tenseurs sticks* lorsque seule une de leurs valeurs propres est non-nulle, et *tenseurs boules* lorsqu'elles sont toutes deux égales à un.

4.1.2 Vote des vecteurs vitesse

Une première méthode de segmentation du mouvement image à l'aide des tenseurs symétriques du second ordre, est proposée dans [59]. Les tenseurs y sont employés pour coder le déplacement visuel, en tout point de l'image, dans l'espace et le temps. L'information sur le mouvement apparent est ensuite propagée, au voisinage de chaque tenseur, à l'aide d'un processus de diffusion isotrope appelé *vote*. La configuration spatiale du tenseur résultant de l'accumulation des votes en chaque site, ou pixel de l'image, permet enfin de déterminer les zones de discontinuité du flot optique.

Codage de l'information

Soit $\omega = (\omega_x, \omega_y)^T$ le vecteur de déplacement mesuré en un point quelconque de l'image, entre deux acquisitions successives. En incluant la dimension temporelle, soit le temps $\triangle_t$ séparant deux prises de vue, on obtient alors :

$$\mathbf{v} = (\omega_x, \omega_y, \triangle_t)^T \ . \tag{4.5}$$

Toutefois, puisque $\triangle_t$ est constant en tout point de l'image, l'équation (4.5) peut être simplifiée de la manière suivante :

$$\mathbf{v} = (\omega_x, \omega_y, 1)^T \ ,$$

sans perte d'information sur la continuité du mouvement spatio-temporel.

La représentation graphique d'un tenseur générique 3-D est une ellipsoïde (Fig. 4.2) définie par le système propre lié à la représentation matricielle $\mathbf{T}$:

$$\mathbf{T} = \left[\begin{array}{cc} \hat{\mathbf{e}}_1 & \hat{\mathbf{e}}_2 \end{array} \right] \left[\begin{array}{ccc} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{array} \right] \left[\begin{array}{c} \hat{\mathbf{e}}_1^T \\ \hat{\mathbf{e}}_2^T \end{array} \right] .$$

Le déplacement image est codé à l'aide d'un tenseur *stick* de dimension trois. Le vecteur $\mathbf{v}$ décrit ainsi l'unique demi-diamètre non nul de l'ellipsoïde :

$$\lambda_1 \hat{\mathbf{e}}_1 = \mathbf{v}, \lambda_2 = \lambda_3 = 0, \tag{4.6}$$

et :

$$\mathbf{T} = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T \ .$$

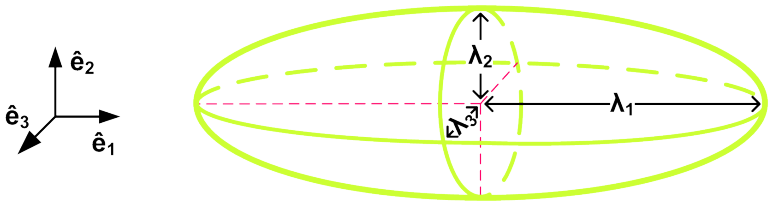


FIG. 4.2 – Représentation graphique d'un tenseur 3-D dans dans son repère propre.

L'utilisation d'une composante temporelle, bien que constante, prend ici tout son sens. Elle prévient l'ambiguïté causée par la représentation tensorielle 2-D des vecteurs spatialement opposés ω et $\omega' = -\omega$, pour lesquels :

$$\lambda_1 \hat{\mathbf{e}}'_1 = \omega' = \lambda_1 (-\hat{\mathbf{e}}_1)$$

d'où :

$$\mathbf{T}' = \lambda_1 (-\hat{\mathbf{e}}_1)(-\hat{\mathbf{e}}_1)^T = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T = \mathbf{T}.$$

L'ajout d'une troisième composante égale à 1 permet d'assurer l'unicité de la représentation tensorielle. Les vecteurs vitesse sont ensuite normalisés avant l'étape de codage, afin de préserver leur équité respective, avec $\lambda_1 \hat{\mathbf{e}}_1 = \hat{\mathbf{v}}$.

Processus de vote

Le vote est un processus d'accumulation au cours duquel chaque point de l'image transmet, à son voisinage, l'information tensorielle qui caractérise son déplacement apparent. Cette information est propagée de manière isotrope à l'aide d'une fonction gaussienne, modélisant l'influence du vote en fonction de la distance s séparant le tenseur votant d'un tenseur cible de son voisinage :

$$W(s) = e^{-\left(\frac{s}{\sigma}\right)^2}, \tag{4.7}$$

avec σ le coefficient d'atténuation. La portée du vote d'un tenseur est bornée spatialement, de sorte que l'information transmise, à savoir un tenseur identique pondéré par la fonction d'atténuation $W(\cdot)$, ait une taille supérieure à 10% de celle du tenseur original.

D'après les propriétés de l'algèbre matricielle, le résultat de l'addition de deux tenseurs symétriques du second ordre, est un tenseur de même nature. L'accumulation des tenseurs durant la phase de vote aboutit donc à un tenseur générique 3-D.

Soit $T_{p,1}$ le tenseur au site p, le résultat du vote, en ce site, est donné par :

$$T_{p,2} = T_{p,1} + \sum_{i}^{\Omega} W\left(\|\mathbf{x}_i - \mathbf{x}_p\|\right) T_{i,1}\,,$$

où $T_{i,1}$ décrit l'ensemble des tenseurs sur le voisinage Ω de p. L'axe principal $\lambda_1'\hat{\mathbf{e}}_1' = (a,b,c)^T$ du tenseur générique $T_{p,2}$ offre une mesure lissée ω' du flot optique, équivalant à la moyenne pondérée à l'aide de la fonction gaussienne $W(\cdot)$ du mouvement apparent calculé sur Ω :

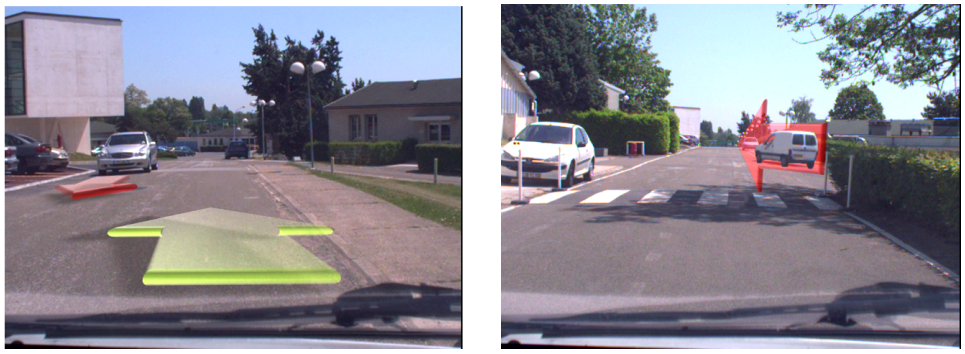
$$\omega' = \left(\frac{a}{c}, \frac{b}{c}\right)^T .$$

La somme de deux tenseurs *sticks* colinéaires est un tenseur de même direction. A l'inverse, plus les tenseurs sommés tendent à être orthogonaux, plus le tenseur résultant est isotrope. La géométrie de $T_{p,2}$ permet donc de mesurer l'hétérogénéité, de direction et de norme, du flot optique au voisinage du site p. On réalise cette mesure d'hétérogénéité par le rapport des axes principaux du tenseur $T_{p,2}$ en chaque site :

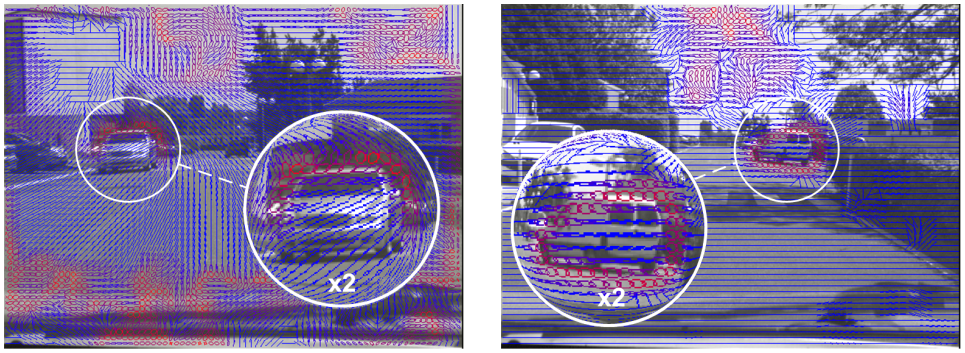
$$\varepsilon = \frac{\lambda_2}{\lambda_1} \tag{4.8}$$

Le résultat de cette mesure dans différents cas est illustré Fig. 4.3. L'accumulation des tenseurs *sticks* non-colinéaires, aux frontières des régions ayant un mouvement relatif non nul, révèle le contour des ensembles de points à segmenter. La représentation dense de l'indice d'hétérogénéité (4.8) offre ainsi une pré-segmentation du flot optique. Elle souligne dans le même temps l'incohérence du flot optique calculé sur les zones de texture uniforme et les zones d'occultation.

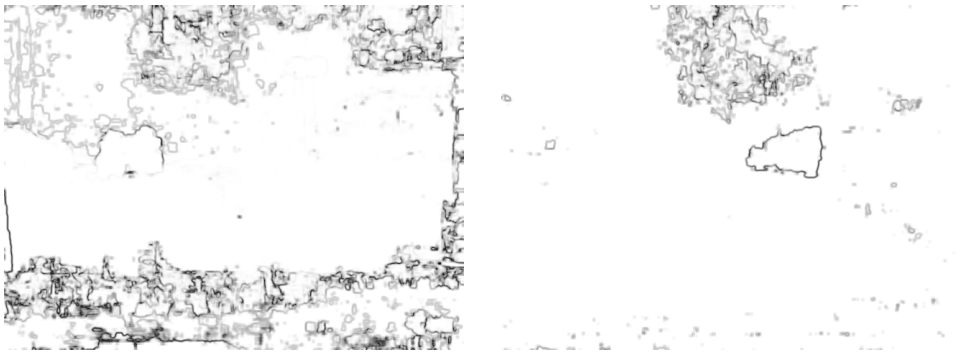
Toutefois, certaines discontinuités du flot optique ne sont pas clairement identifiées par cette méthode, notamment lorsqu'elles correspondent à une rupture sur la norme des vecteurs vitesse, tandis que la composante temporelle (unitaire) est négligeable par rapport au mouvement. Ainsi, le fossé situé à droite de l'espace navigable et la zone de contact entre la route et le véhicule ne sont pas clairement identifiés dans l'exemple du capteur mobile de la figure 4.3.



Deux acquisitions vidéo. En vert, le déplacement de la caméra embarquée. En rouge, celui des obstacles mobiles observés.



Champs de tenseurs projetés dans le plan image à l'issue du vote. Du bleu au rouge, dans le sens croissant de la mesure d'hétérogénéité du flot optique.



Carte dense de la mesure d'hétérogénéité du flot optique. Du blanc au noir, dans le sens croissant des valeurs.

FIG. 4.3 – Illustration du processus de vote dans le cas d'acquisitions réalisées par un capteur mobile (à gauche) ou statique (à droite).

4.2 Tensor Voting

4.2.1 Formalisme

Le *Tensor Voting* est une approche initialement développée par G. Medioni, pour la perception des structures géométriques formées par l'agencement des points d'un espace n-D. Il s'appuie sur la loi de continuité, extraite des principes d'organisation perceptuelle de Gestalt [63], qui tentent de définir le fonctionnement de la perception visuelle humaine. Selon ces principes, le détail de chaque élément constitutif d'un ensemble est perçu *a posteriori* de ce dernier. Formulé différemment, les points d'un espace sont instinctivement regroupés d'après leur distance relative, de manière à identifier les structures qu'ils dessinent. Le *Tensor Voting* tend à simuler ce procédé, à l'aide d'une représentation tensorielle des structures géométriques et d'un processus de diffusion servant à propager l'information structurelle au voisinage de chaque point.

La décomposition (4.3) d'un tenseur générique n-D, met en évidence n tenseurs élémentaires distincts, d'ordre croissant jusqu'à n. Chaque tenseur élémentaire est associé à l'une des n structures géométriques d'un espace n-dimensionnel, de sorte que le nombre de normales de cette structure correspond à la dimension du tenseur considéré. On obtient de cette façon une représentation tensorielle bijective, permettant de définir chaque structure et son orientation, en tout point de l'espace.

Pour détailler les étapes du *Tensor Voting*, on étudie précisement le cas d'un espace 2-D, dans lequel les tenseurs élémentaires, au nombre de deux, sont obtenus par la décomposition (4.3) de la forme quadratique $\mathbf{T}$ d'un tenseur générique :

$$\mathbf{T} = (\lambda_1 - \lambda_2)\underbrace{\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T}_{stick} + \lambda_2\underbrace{\left(\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T\right)}_{boule}. \tag{4.9}$$

Les tenseurs isotropes de type *boule* désignent les points isolés ou localisés au sein d'une région, de sorte qu'aucune orientation prédominante ne caractérise localement la structure à laquelle ils appartiennent. Au contraire, les *tenseurs sticks* sont utilisés pour coder les structures curvilignes, de façon à représenter la normale à ces dernières au point considéré. Toutefois, comme aucune information ne permet d'identifier ces structures à l'initialisation du *Tensor Voting*, tous les points sont dans un premier temps, codés par des tenseurs unitaires de type *boule* (Fig. 4.4(a)). L'étape de diffusion permet alors de spécialiser les tenseurs T_p, grâce à l'accumu-

lation des votes $V(T_i)$ émis sur leur voisinage Ω :

$$T_{p,2} = T_{p,1} + \sum_{i}^{\Omega} V(T_{i,1}) .$$

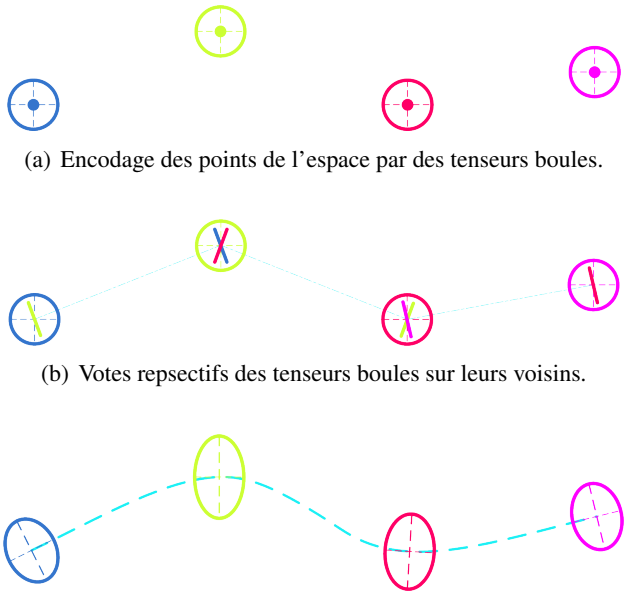
A l'aide de la représentation tensorielle précédemment décrite, chaque vote propage au tenseur cible, l'information structurelle du lien connectant ce dernier au tenseur votant. En l'absence de toute autre indication, la ligne droite est le moyen le plus simple et direct, pour relier deux ensembles isotropes d'un espace. Le vote d'un tenseur *boule* se résume donc à un tenseur *stick,* orthonormal au vecteur $\mathbf{v}$ joignant le site cible (Fig. 4.4(b)). Sa forme quadratique est obtenue en retranchant le produit direct du vecteur $\widehat{\mathbf{v}}$ (*i.e.* le tenseur stick unitaire tangent à $\mathbf{v}$), de la matrice identité 3×3 correspondant au tenseur unitaire de type *boule*. Et puisque la probabilité que deux points appartiennent à une même structure géométrique diminue avec la distance les séparant, l'influence d'un tenseur doit décroître de la même manière. On utilise à cet effet une fonction de décroissance sur $\mathbf{v}$, similaire à la fonction d'atténuation (4.7), modélisée par une gaussienne centrée sur la position du tenseur votant :

$$\mathbf{V} = e^{-\left(\frac{\|\mathbf{v}\|}{\sigma}\right)^2} \left(\mathbf{I} - \widehat{\mathbf{v}}\widehat{\mathbf{v}}^T\right) . \tag{4.10}$$

A l'issue du vote, la décomposition des tenseurs spécialisés, selon l'équation (4.9), permet de déterminer le type de structure géométrique que forme chaque point de l'espace étudié avec son voisinage :

- Lorsque $\lambda_1 - \lambda_2 > \lambda_2$, le terme correspondant au tenseur élémentaire de type *stick* prédomine. La structure au point considéré est donc curviligne, de normale orientée selon $\hat{\mathbf{e}}_1$.
- Pour $\lambda_1 \approx \lambda_2 > 0$, le tenseur obtenu est isotrope. Il définit le point d'intersection de plusieurs courbes, ou son appartenance à une région.
- Enfin, les points isolés n'appartenant à aucune structure géométrique sont identifiables par leurs faibles valeurs propres.

La figure 4.4 résume les différentes étapes du processus permettant d'identifier une courbe passant par quatre points voisins d'un espace 2-D. Le principe est identique pour les espaces de dimension supérieure. En 3-D par exemple, la surface est ajoutée à la liste des structures géométriques étudiées. La représentation tensorielle de ces dernières est établie d'après la règle énoncée en début de section : les surfaces



(a) Encodage des points de l'espace par des tenseurs boules.

(b) Votes repsectifs des tenseurs boules sur leurs voisins.

(c) Résultat du vote : la courbe perçue est définie suite à la spécialisation de chaque tenseurs.

FIG. 4.4 – Détail du processus de vote 2-D sur des données isotropes.

sont représentées par leurs normales, codées à l'aide des tenseurs *sticks* ($\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T$) ; les courbes 3-D sont définies par des tenseurs *disques* ($\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T$), placés orthogonalement à leur tangente en chaque point ; les points sont encodés par des tenseurs de type *boule* ($\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T + \hat{\mathbf{e}}_3\hat{\mathbf{e}}_3^T$). Enfin, l'équation (4.10), décrivant le vote d'un tenseur *boule* en son voisinage, reste valable quelle que soit la dimension considérée.

Le champ d'action des tenseurs durant le processus de vote est l'unique paramètre à régler, à travers la détermination de l'écart-type σ dans la fonction de décroissance de l'équation 4.10. Par ailleurs l'algorithme est intégralement parallélisable, puisque chaque point de l'espace étudié est traité indépendamment des autres. La complexité algorithmique du *Tensor Voting* sur une architecture parallèle peut donc être en $O(n)$, où n désigne le nombre de points considérés.

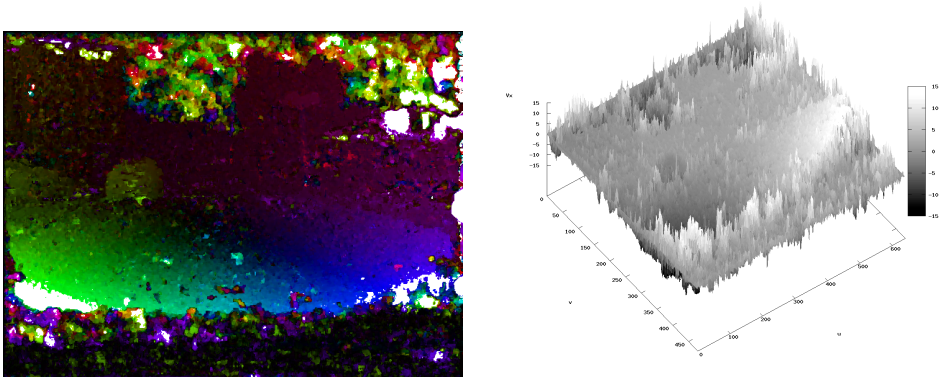
4.2.2 *Tensor Voting* et flot optique

L'espace (x, y, v_x, v_y)

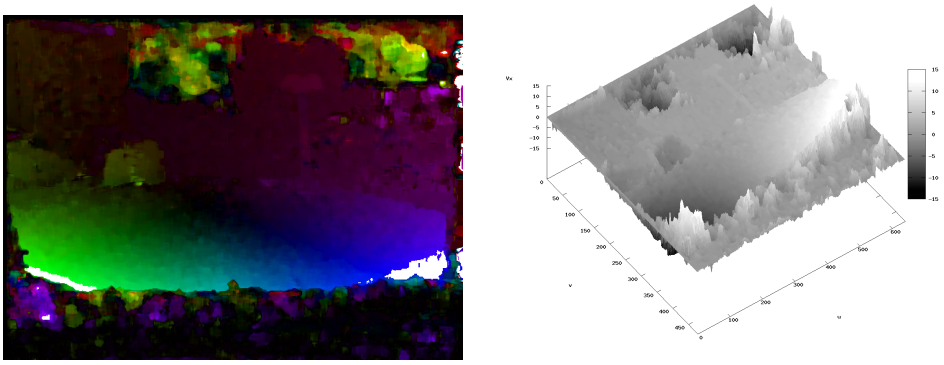
Soient (v_x, v_y) les composantes du vecteur vitesse au point de coordonnées (x, y). L'analyse des discontinuités du mouvement image est similaire à l'étude

de certaines discontinuités structurelles de l'espace 4-D (x, y, v_x, v_y). Dans cet espace, le champ de déplacement visuel peut former plusieurs hypersurfaces 4-D, correspondant chacune à une région connexe maximale de l'image, de mouvement apparent homogène (Fig. 4.5). On entend par homogène, continu à la discrétisation près du domaine spatial de définition de l'image. De façon duale, le contour de ces surfaces désigne les discontinuités recherchées du mouvement image. Différentes causes peuvent être à l'origine des ruptures de continuité du champ de vecteurs vitesse :

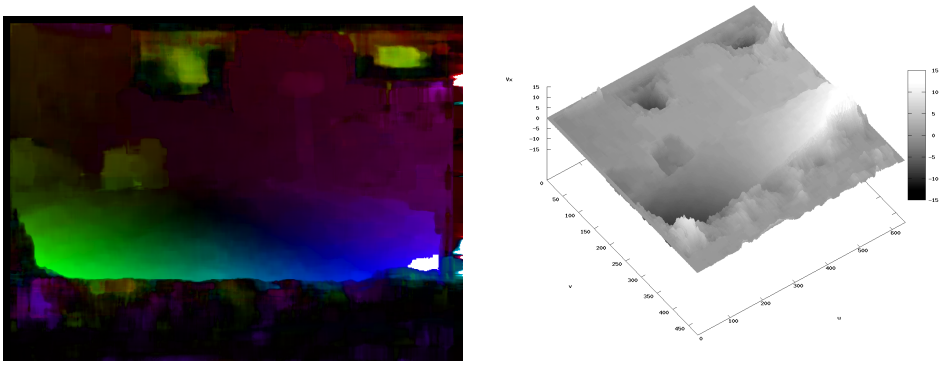
- D'une part, au sein d'un environnement statique, elles sont principalement dues à la parallaxe de mouvement, c'est-à-dire à l'incidence du changement de position de l'observateur sur l'observation d'un objet. Lorsqu'une translation est appliquée à la caméra, la position relative des objets dans l'image varie en fonction de leur profondeur relative dans la scène. A l'inverse, une rotation pure de la caméra n'induit aucune parallaxe, puisqu'elle ne modifie la profondeur relative des éléments de la scène. Dans le cadre des Systèmes de Transport Intelligents, on admet un modèle cinématique non-holonomique de type "bicyclette", de sorte qu'il est impossible d'avoir une composante en translation nulle lors du déplacement du véhicule.
- D'autre part, en situation réelle et notamment en milieu urbain où la concentration d'individus est importante, les discontinuités du mouvement image sont induites par l'observation d'éléments mobiles. Toutefois, si dans la majorité des cas, le déplacement apparent des objets en mouvement diffère du flot optique des points de l'arrière plan, il peut arriver que le déplacement apparent de ces obstacles soit consistant avec un objet statique, et donc interprétés comme tel. Le chapitre 6 revient plus en détails sur ce problème.
- Enfin, l'analyse du flot optique doit prendre en compte les erreurs d'estimation du mouvement image. En fonction de l'algorithme employé, le champ de vecteurs vitesse calculé peut être ou non régularisé, donc plus ou moins lissé et donc enclin aux ruptures de continuité, à la frontière des régions dont le mouvement relatif est non nul. Il est important de privilégier l'utilisation d'une approche locale, faiblement régularisée, de manière à pouvoir identifier toute mauvaise estimation comme une aberration structurelle dans l'espace (x, y, v_x, v_y). Ainsi, et à l'inverse des approche variationnelles globales (section 3.2.3) qui comportent un terme de régularisation, l'implémentation pyramidale de l'algorithme de Lucas et Kanade satisfait cette requête.



$\Omega_{ROI} = 4 \times 4$



$\Omega_{ROI} = 10 \times 10$



$\Omega_{ROI} = 20 \times 20$

FIG. 4.5 – Visualisation du sous-espace (x, y, v_x) (colonne de droite) correspondant au flot optique mesuré pour différentes ouvertures (colonne de gauche).

***Tensor Voting* 4-D**

L'emploi du *Tensor Voting* au sein de l'espace (x, y, v_x, v_y) permet de retrouver les hypersurfaces 4-D correspondant aux régions de l'image pour lesquelles le flot optique respecte une certaine continuité. Dans un premier temps, les points de l'image sont encodés par des tenseurs 4-D unitaires de type *boule*. L'agencement relatif des points de l'espace 4-D est ensuite diffusé par l'intégralité des tenseurs, à l'aide du processus de vote présenté dans la section 4.2.1 et décrit par la formule (4.10). Les tenseurs résultants sont ensuite décomposés en tenseurs élémentaires :

$$\begin{aligned} \mathbf{T} &= (\lambda_1-\lambda_2)\underbrace{\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T}_{stick} + (\lambda_2-\lambda_3)\underbrace{\left(\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T+\hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T\right)}_{disque\,2D} \\ &+ (\lambda_3-\lambda_4)\underbrace{\left(\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T+\hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T+\hat{\mathbf{e}}_3\hat{\mathbf{e}}_3^T\right)}_{disque\,3D} \\ &+ \lambda_4\underbrace{\left(\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T+\hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T+\hat{\mathbf{e}}_3\hat{\mathbf{e}}_3^T+\hat{\mathbf{e}}_4\hat{\mathbf{e}}_4^T\right)}_{boule} \end{aligned} \tag{4.11}$$

Une hypersurface 4-D est définie en tout point de l'espace considéré par deux normales. Sa représentation tensorielle correspond donc au disque de dimension deux, codé par $\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T$. A l'issue du vote, on utilise donc la valeur du coefficient $\lambda_2 - \lambda_3$ de chaque tenseur comme indice de continuité du flot optique au point de coordonnées (x, y) dans le plan image.

Outre la complétude du formalisme employé, l'avantage de cette approche réside dans l'incidence de la topographie des points de l'espace (x, y, v_x, v_y), sur le champ d'action des tenseurs dans le plan (x, y). La distance induite par la hauteur séparant, dans cet espace $4-D$, deux régions pourtant connexes dans le plan image, limite le vote des tenseurs des points d'une régions sur ceux de l'autre, aux abords du contour des hypersurfaces.

Toutefois, le temps d'exécution du *Tensor Voting 4*-D appliqué à un champ de déplacement de 640×480 éléments, est approximativement de vingt secondes sur une architecture mono-core[1], lorsque σ est égal à quatre. Cela dépasse de beaucoup le délai de décision admissible en robotique mobile, de l'ordre de quelques millisecondes. Aussi propose-t-on au chapitre 7, une solution originale, basée sur l'utilisation d'un GPU et l'interface de programmation CUDA, permettant de satisfaire la contrainte temps-réel.

[1] Processeur Intel Pentium 4 de 3,2 GHz.

Normalisation du résultat

Pour la suite des travaux, il est utile de normaliser le coefficient $\lambda_2 - \lambda_3$, appliqué au tenseur élémentaire codant les points d'une hypersurface 4-D. La valeur maximale est obtenue dans le cas d'un tenseur inclut dans un hyperplan 4-D, pour lequel les composantes v_x et v_y sont constantes en tout point. Soit O, un point de cet hyperplan, et P_i, $i \in \{1;2;3;4\}$, quatre points voisins situés à égale distance $s = \|\mathbf{v}\|$ de O, positionnés comme illustré dans la figure 4.6. Les votes $\mathbf{V}_i$, au point O, des tenseurs associés aux points P_i, sont données par l'équation (4.10) :

$$\mathbf{V}_1 = \mathbf{V}_3 = e^{-(s^2/\sigma^2)} \begin{bmatrix} 1-Vx^2 & -VxVy & 0 & 0 \\ -VxVy & 1-Vy^2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{V}_2 = \mathbf{V}_4 = e^{-(s^2/\sigma^2)} \begin{bmatrix} 1-Vy^2 & VxVy & 0 & 0 \\ VxVy & 1-Vx^2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

avec $\mathbf{v} = (Vx, Vy)^T$, d'où :

$$\mathbf{V}_1 + \mathbf{V}_2 + \mathbf{V}_3 + \mathbf{V}_4 = 4\,e^{-(s^2/\sigma^2)} \begin{bmatrix} 1-\left(\frac{Vx^2+Vy^2}{2}\right) & 0 & 0 & 0 \\ 0 & 1-\left(\frac{Vx^2+Vy^2}{2}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

La somme des votes des tenseurs du voisinage Ω, autour du point O, ajoutée au tenseur boule codant ce dernier, s'écrit donc :

$$\mathbf{T}_{max} = 4\sum_{4}^{\Omega} e^{-(s^2/\sigma^2)} \begin{bmatrix} 1-\left(\frac{Vx^2+Vy^2}{2}\right) & 0 & 0 & 0 \\ 0 & 1-\left(\frac{Vx^2+Vy^2}{2}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Les valeurs propres de $\mathbf{T}_{max}$ sont directement lisibles et ordonnables à partir de

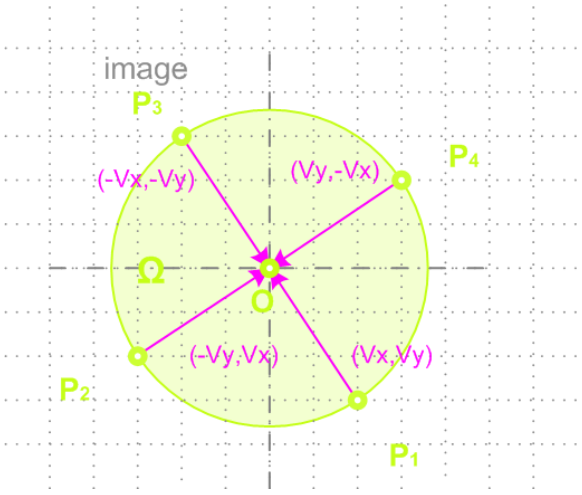


FIG. 4.6 – Projection d'un hyperplan 4-D dans le plan image.

cette écriture :

$$\lambda_1 = \lambda_2 = \sum^{\frac{\Omega}{4}} 8\, e^{-\left(s^2/\sigma^2\right)}$$
$$\lambda_3 = \lambda_4 = \sum^{\frac{\Omega}{4}} 4\, e^{-\left(s^2/\sigma^2\right)} \left(2 - \left(\frac{Vx^2+Vy^2}{2}\right)\right)$$

Les coefficients associés au tenseur élémentaire codant les hypersurfaces 4-D sont donc bornés par la valeur maximale :

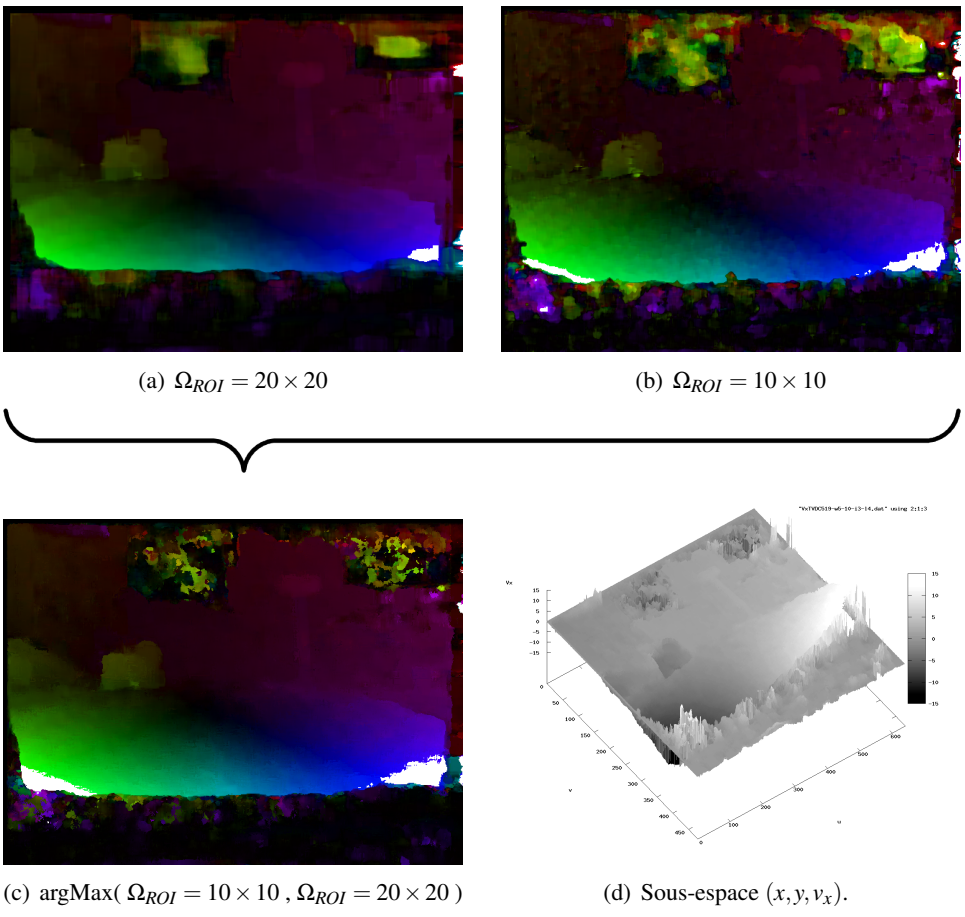
$$\lambda_2 - \lambda_3 = \sum^{\frac{\Omega}{4}} 2\, e^{-\left(s^2/\sigma^2\right)} \left(Vx^2 + Vy^2\right).$$

Vote combiné

G. Medioni emploie pour la première fois le *Tensor Voting* associé à l'espace joint des coordonnées image et du mouvement apparent [60] pour segmenter les objets mobiles observés par une caméra statique. Il estime alors le mouvement image à l'aide d'un algorithme de corrélation par bloc et du *Tensor Voting*. Pour chaque pixel, la mesure du déplacement est choisie parmi les trois meilleurs résultats de la fonction de corrélation, en fonction de la valeur $\lambda_2 - \lambda_3$ du tenseur 4-D correspondant. Plus grande est cette valeur, plus régulier sera le champ de vecteurs vitesse autour du pixel considéré.

La version pyramidale de l'algorithme de Lucas et Kanade permet de calculer des déplacements importants tout en limitant l'ouverture utilisée pour contraindre

l'équation du flot optique. Toutefois, l'estimation réalisée aux plus basses résolutions (au sommet de la pyramide) peut faire diverger le processus itératif lorsque le sous-échantillonnage du signal détériore trop fortement les images. Le raffinement temporel semble donc être une solution cohérente pour corriger ces erreurs, en initialisant le processus d'estimation du flot optique à l'aide du déplacement calculé pour la paire d'images précédente. Cependant le calcul du mouvement apparent doit alors être exécuté sur un seul niveau pour ne pas diverger de façon identique à l'approche pyramidale classique. Pour combiner les avantages d'une approche pyramidale et du raffinement temporel (section 3.1.2), on se propose de suivre une démarche similaire à celle proposée par G. Medioni. Ainsi, le *Tensor Voting* est appliqué simultanément à deux champs de déplacement, l'un issu d'une approche pyramidale, et l'autre, obtenu par raffinement temporel.



(a) $\Omega_{ROI} = 20 \times 20$ (b) $\Omega_{ROI} = 10 \times 10$

(c) argMax($\Omega_{ROI} = 10 \times 10$, $\Omega_{ROI} = 20 \times 20$) (d) Sous-espace (x, y, v_x).

FIG. 4.7 – Résultat de la combinaison par Tensor Voting (c) de deux champs de déplacement (a, b) et la représentation du sous-espace (x, y, v_x) correspondant (d).

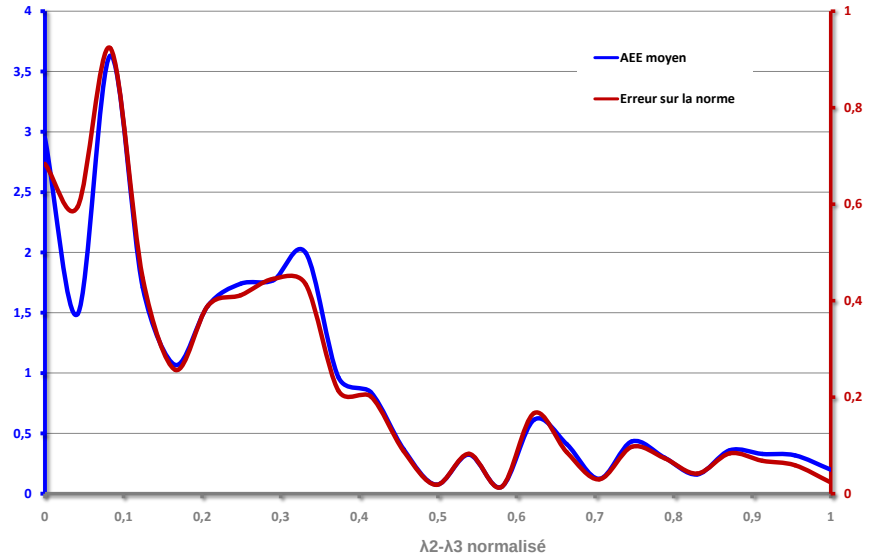


FIG. 4.8 – Erreur moyenne, angulaire (en bleu) et sur la norme (en rouge), du flot optique en fonction de la valeur normalisée $\lambda_2 - \lambda_3$ des tenseurs de l'image (séquence témoin *Yosemite*).

On regroupe donc les jeux de points 4-D correspondant à l'estimation des deux champs de déplacement précédemment décrits et illustrés Fig. 4.7(a) et (b). Ainsi, lors du *Tensor Voting*, en chaque coordonnée (x,y) de la matrice image, deux tenseurs, correspondant chacun à une estimation flot optique, reçoivent les votes de leurs voisinages respectifs. A l'issue du vote, l'estimation finale du déplacement apparent, en ce point, est donné par celui des deux qui possède plus fort coefficient $\lambda_2 - \lambda_3$. Le *Tensor Voting* est alors ré-employé sur le flot optique "combiné", afin de déterminer les discontinuités de ce dernier. La figure 4.7 illustre un exemple de combinaison de deux champs de déplacement dont les estimations utilisent une ouverture respective de taille 10×10 et 20×20 pixels. Dans cet exemple, la continuité globale du champ de déplacement final est assurée par l'utilisation d'une ouverture large (20×20) et la précision des contours du mouvement apparent par une ouverture de taille plus réduite 10×10.

Enfin, on réalise une étude quantitative sur l'erreur d'estimation du flot optique en fonction de la valeur du coefficient associé au tenseur élémentaire codant les hypersurfaces 4-D. On mesure donc sur une séquence synthétique, l'erreur moyenne du flot optique, en fonction de la valeur normalisée du coefficient $\lambda_2 - \lambda_3$ associé au mouvement apparent des points de l'image. Le résultat sur la séquence *Yose-*

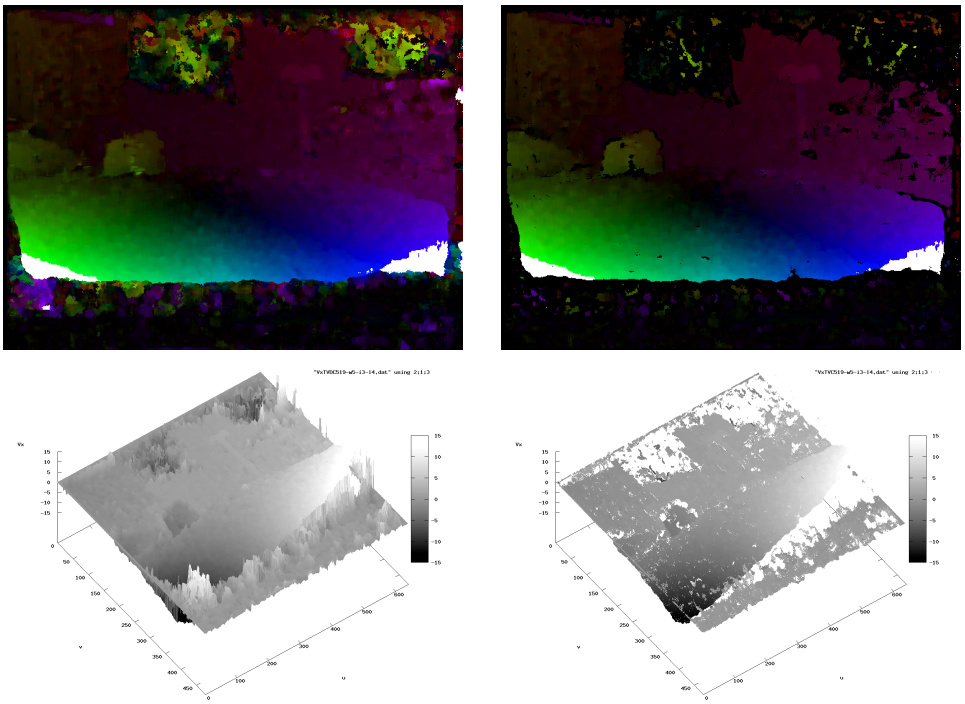


FIG. 4.9 – A gauche, le flot optique combiné par *Tensor Voting*. A droite, le même champ de déplacement, sans les estimations aberrantes correspondant aux valeurs $\lambda_2 - \lambda_3 > 0.5$.

mite est illustré par le graphique Fig. 4.8. On peut y observer qu'au delà d'un seuil $\lambda_2 - \lambda_3 = 0,4$, l'erreur moyenne du déplacement estimé tend à se stabiliser dans une plage de valeurs comprises entre 0 et $0,2$ degré pour l'erreur angulaire, et entre 0 et $0,6$ pixel pour l'erreur sur la norme. Le résultat du *Tensor Voting* offre donc une mesure de confiance du flot optique, permettant de discriminer les vecteurs aberrants du champ de déplacement si nécessaire (Fig. 4.9).

4.3 Segmentation du flot optique

4.3.1 Ligne de partage des eaux

La mesure de confiance, précédemment établie, correspond en réalité à une mesure de continuité du flot optique. Aussi l'image des coefficients normalisés $\lambda_2 - \lambda_3$ s'apparente-t-elle à une carte des discontinuités du mouvement apparent, représentant le contour des régions homogènes du champ de déplacement estimé. La segmenter revient donc à segmenter le flot optique.

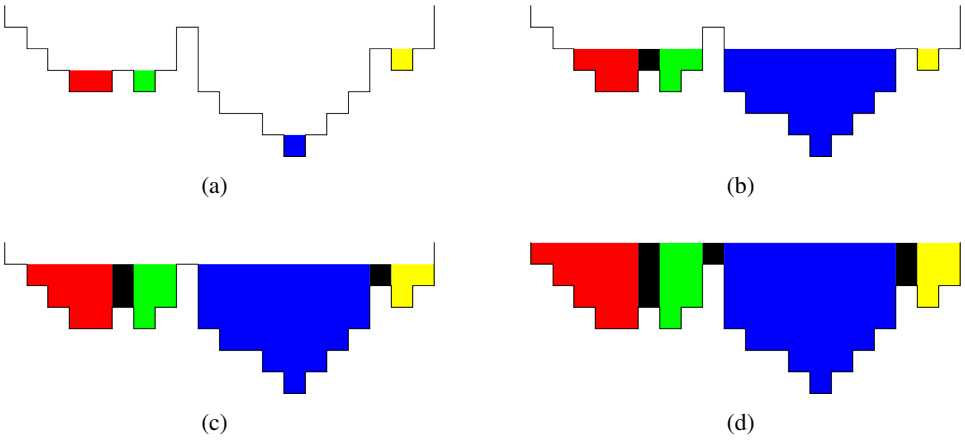


FIG. 4.10 – Identification des lignes de partage des eaux dans un signal 1-D.

L'obtention de contours fermés, sur un seul niveau d'intensité, est obtenu à l'aide d'un algorithme de segmentation par ligne de partage des eaux [75, 76], qui considère une image en niveaux de gris comme un relief topographique dont on simule l'inondation. Le relief est immergé progressivement dans l'eau, qui pénètre en différents points ou marqueurs, tels que les minima locaux de l'image. L'inondation se poursuit de manière à propager les bassins formés sur l'intégralité du relief. Lorsque deux bassins se rejoignent au niveau d'une ligne de crête, cette dernière est surélevée (en noir sur le schéma 4.10) de manière à prévenir la fusion des bassins. Une fois le relief totalement immergé, la segmentation de l'image originale est donnée par ces lignes de partage des eaux, ou *watersheds*, qui correspondent aux contours de chaque région associée à un marqueur de minima local.

Algorithm 1 LPE : Algorithme de F. Meyer

ENTRÉE(S): Image en niveaux de gris
SORTIE(S): Lignes de partage des eaux

1: Marquer les minimas locaux de l'image.
2: **TANT QUE** chaque pixel n'est pas marqué **FAIRE**
3: sélectionner le pixel non marqué de moindre intensité, adjacent à un bassin.
4: **SI** il est connexe à exactement 1 bassin **ALORS**
5: l'inclure au bassin considéré.
6: **SINON**
7: l'identifier comme appartenant à une ligne de partage des eaux.

Pour simuler la progression des bassins, il est nécessaire de définir la notion d'adjacence entre les pixels d'une image. Soit $\mathbb{Z}$ l'ensemble des entiers, on note $\mathbb{I}^2$

le produit cartésien des deux sous-ensembles de $\mathbb{Z}$, $\{0,\cdots,w-1\}$ et $\{0,\cdots,h-1\}$, pour lesquels w et h représentent respectivement la largeur et la hauteur d'une image. Les relations d'adjacence généralement considérées dans le cas d'un maillage rectangulaire sont la 4 et la 8-connexité, définies pour tout point $\mathbf{x} = (u,v) \in \mathbb{I}^2$, par :

$$\begin{aligned} \Gamma_4(\mathbf{x}) &= \left\{\mathbf{x}_i \in \mathbb{I}^2; |u-u_i| + |v-v_i| \leq 1\right\} \\ \Gamma_8(\mathbf{x}) &= \left\{\mathbf{x}_i \in \mathbb{I}^2; \max\left(|u-u_i|, |v-v_i|\right) \leq 1\right\} \end{aligned}$$

Deux points $\mathbf{x}_1$ et $\mathbf{x}_2$, $\mathbf{x}_1 \neq \mathbf{x}_2$, sont dits *n-connexes* si $\mathbf{x}_2 \in \Gamma_n(\mathbf{x}_1)$.

Les algorithmes efficaces de segmentation par Ligne de Partage des Eaux (LPE), tels que celui proposé par F. Meyer, ont une complexité linéaire selon le nombre de pixels de l'image, les classant parmi les méthodes de segmentation les plus rapides. En pratique, ils sont rarement appliqués à l'image originale. Une étape préalable de filtrage est en effet nécessaire pour éviter la sur-segmentation induite par le nombre élevé de minima locaux au sein du signal original. Dans le cadre des travaux qui sont présentés, on utilise à cet effet les *opérateurs morpho-mathématiques connexes* étudiés dans la section suivante avant d'être appliqués à l'image des coefficients normalisés $\lambda_2 - \lambda_3$.

4.3.2 Opérateurs connexes et filtrage par attribut

Le filtrage a généralement pour but d'éliminer l'information non pertinente d'un signal tout en préservant l'intégralité de l'information utile. Dans le cas du pré-traitement de l'image, avant sa segmentation par Ligne de Partage des Eaux (LPE), la pertinence cette information est fonction de critères prédéfinis sur les bassins finaux (leur nombre, leur taille, leur géométrie, etc.). Ainsi, par exemple, si l'objectif est de limiter le nombre de régions obtenues à l'issue de la segmentation, il faut alors, en amont, limiter d'autant le nombre de minimas locaux (les marqueurs). En pratique, la définition d'un tel opérateur de filtrage est un problème complexe. Les contours du signal sont généralement modifiés, notamment par le lissage des filtres linéaires et gaussiens ou par l'emploi d'opérateurs morphologiques à base d'éléments structurants fixes. Les morpho-mathématiques proposent cependant un certain nombre d'opérateurs, dits connexes, dont l'une des propriétés est la préservation des contours du signal original.

Les opérateurs connexes apparaissent pour la première fois en 1976 [77], avec *l'ouverture par reconstruction d'un signal binaire*, définie comme l'élimination des seules composantes connexes intégralement supprimées par l'érosion à l'aide

d'un élément structurant donné. Une définition formelle des opérateurs binaires connexes est apportée dans [78, 79] :

Définition 1 *Un opérateur binaire ψ est connexe lorsque pour toute image X, l'ensemble $E \setminus \psi(X)$ est exclusivement composé de composantes connexes de X ou de son complément X^c.*

La généralisation aux opérateurs sur des images en niveaux de gris est liée au concept de *partition* :

Définition 2 *Une partition d'un espace E est un ensemble de composantes connexes $\{A_i\}$ disjointes ($A_i \cap A_j = \varnothing, i \neq j$) dont l'union correspond à l'espace entier. Chaque composante A_i est appelée* une classe *de la partition.*

Une partition $\{A_i\}$ est dite plus *fine* qu'une autre partition $\{B_i\}$ si chaque classe de $\{A_i\}$ est incluse dans une classe de $\{B_i\}$. La *partition associée* d'une image binaire X contient les composantes connexes de X et leur complément. Un opérateur binaire ψ est donc connexe, si et seulement si, pour toute image X, la partition associée de cette image est plus fine que la partition associée de $\psi(X)$. Il n'agit ainsi que par suppression de composantes connexes, de l'image ou du fond. Par ailleurs, la partition associée d'une image en niveaux de gris correspond à l'ensemble des composantes connexes maximales de même intensité. P. Salembier et J. Serra étendent la définition des opérateurs connexes aux *fonctions en niveaux de gris*, à l'aide de la définition suivante :

Définition 3 *Un opérateur ψ, appliqué à une fonction en niveau de gris est dit connexe si, pour toute fonction f, la partition associée de $\psi(f)$ est moins fine que la partition associée de f.*

En associant à une fonction la partition des régions connexes maximales de même intensité, un opérateur connexe permet de simplifier une image en fusionnant certaines de ces régions.

La partition associée d'une image en niveaux de gris peut être codée à l'aide d'une représentation hiérarchique : *l'arbre des composantes*. Chaque noeud décrit alors une composante connexe et chaque arrête la relation d'adjacence entre deux composantes de niveaux successifs. On différencie toutefois deux représentations hiérarchiques duales appelées *Max* et *Min-Tree*, selon que leurs feuilles correspondent respectivement aux maxima ou aux minima locaux des images.

Algorithm 2 Construction d'un Min-Tree

ENTRÉE(S): Image en niveaux de gris
SORTIE(S): Min-Tree

```
1: POUR chaque pixel par ordre croissant FAIRE
2:    créer un nouveau noeud.
3:    POUR chaque pixel adjacent préalablement traité FAIRE
4:       SI il est de même intensité ALORS
5:          fusionner le noeud courant au noeud associé au pixel voisin.
6:       SINON
7:          ajouter le noeud associé au pixel voisin à la liste des fils du noeud
            courant.
```

Tout opérateur agissant sur l'arbre des composantes par élagage est un opérateur connexe. Le processus de filtrage explore l'arbre et évalue, pour chaque noeud parcouru, un critère spécifique en fonction duquel le noeud est conservé ou supprimé avec le sous-arbre dont il est la racine. On parle alors également de *filtrage par attribut*. Le critère employé, noté $\mathscr{M}(\cdot)$, est généralement croissant de sorte que pour tout couple de noeuds C_{h_1} et C_{h_2}, respectivement situés aux niveaux h_1 et h_2 tels que $h_1 \geq h_2$, $\mathscr{M}(h_1) \leq \mathscr{M}(h_2)$. La figure 4.11 illustre le cas du filtrage par attribut, sur un critère de hauteur, de la représentation *Min-Tree* d'un signal $1-D$: tout noeud correspondant à un minima local de hauteur inférieure à deux est supprimé. La segmentation du signal par Ligne de Partage des Eaux s'en trouve modifiée puisque les bassins rouge et vert d'une part, bleu et jaune d'autre part, sont alors fusionnés.

Une grande variété d'opérateurs connexes est présentée dans la littérature [81, 82, 83, 80]. Dans le cadre de la segmentation du flot optique, problème étudié dans ce document, l'objectif est de déterminer les marqueurs nécessaires au partitionnement de l'image des discontinuités obtenues par le *Tensor Voting* (Fig. 4.12(b)). On propose de ne sélectionner que les maxima locaux de taille supérieure à un seuil prédéfini, à l'aide d'un filtrage par attribut pour un critère de surface, appli-

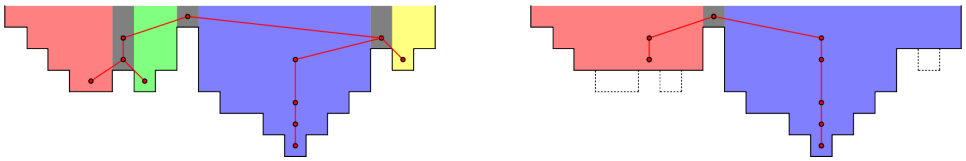


FIG. 4.11 – Filtrage par attribut à l'aide d'un critère de hauteur inférieure à deux, à partir de la représentation *Min-Tree* du signal.

qué au *Max-Tree* de l'image des discontinuités. Les marqueurs sont alors définis par les sous-arbres minimaux dont le nombre de pixels est supérieur au seuil choisi (Fig. 4.12(c)). Le relief topographique correspondant à l'inverse de l'image des discontinuités est ensuite immergé, de façon à identifier les lignes de partage des eaux, dessinées en rouge Fig. 4.12(d). La valeur de seuil du critère est choisie de manière à prévenir la sur-segmentation tout en assurant le partitionnement des régions d'intérêt dans l'image, tels que les obstacles mobiles. Un seuil bas garantit une segmentation fine mais nécessite un effort important pour regrouper ensuite les cellules d'un même objet sémantique. A l'inverse, un seuil haut réduit le nombre de bassins formés, mais augmente le risque de perdre l'information de contour de certains éléments de l'image. Dans l'exemple de la figure 4.12 un seuil d'une valeur de 600 pixels est utilisé, de sorte que les minimas locaux correspondant aux cellules inférieures à 25 pixels carrés ne soient pas utilisés comme marqueurs lors de la segmentation par ligne de partage des eaux.

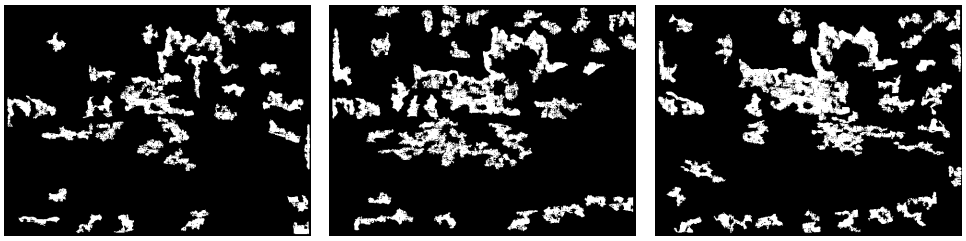
4.4 Conclusions

L'utilisation du *Tensor Voting* dans l'espace joint (x, y, v_x, v_y), offre une méthode élégante de pré-segmentation du flot optique. Grâce aux discontinuités topographiques, dans cet espace, entre certaines régions connexes dans le plan image mais dont le déplacement apparent diffère, le vote de tenseurs devient un processus adaptatif. La différence des valeurs propres λ_2 et λ_3, associée à chaque tenseur de l'espace 4-D, délivre une image en niveaux de gris des discontinuités du flot optique. La segmentation de cette image par Ligne de Partage des Eaux donne alors un partitionnement du mouvement apparent dont la finesse dépend du processus de filtrage préalable.

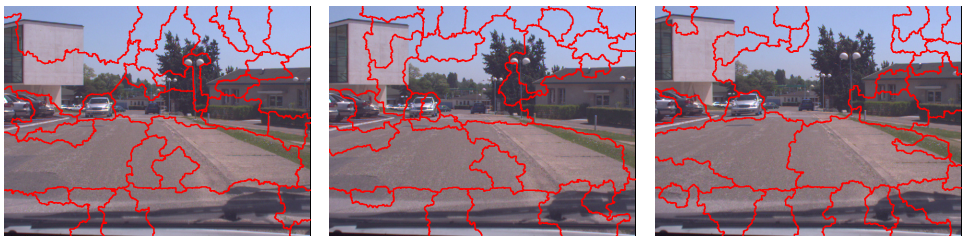
Par ailleurs, le vote combiné sur différents champs de déplacement permet d'utiliser conjointement la pertinence de leur mesure en chaque pixel. On peut ainsi obtenir un flot mixte, formé par la combinaison des meilleures valeurs des flots d'entrée. Le point clef du processus réside dans la capacité du *Tensor Voting* à évaluer le mouvement image estimé, d'après sa cohérence structurelle dans l'espace (x, y, v_x, v_y). Le résultat du *Tensor Voting* apporte donc également un indice de confiance sur le déplacement mesuré. Cet indice est utilisé aux chapitres 5 et 6 pour rendre plus robuste le calcul de certaines contraintes géométriques sur le mouvement. Ces contraintes sont ensuite employées pour classifier les cellules issues de l'étape de segmentation par Ligne de Partage des Eaux.

(a) Images successives de la séquence originale "Route".

(b) Discontinuités du flot optique calculé à l'aide du Tensor Voting 4-D.



(c) Marqueurs obtenus à l'aide d'un filtrage par attribut, pour un critère de type surface, sur l'image des discontinuités.



(d) En rouge, les lignes de jonction des bassins formés à partir des différents marqueurs sélectionnés.

FIG. 4.12 – Segmentation du flot optique par LPE.

Troisième partie

Perception de l'environnement

L'analyse du mouvement apparent permet d'identifier les régions de l'image dont le déplacement relatif est non nul. A cet effet, le chapitre 4 a présenté un exemple de segmentation du flot optique, à l'aide d'un algorithme de segmentation par ligne de partage des eaux appliqué sur l'image des discontinuités obtenue à l'issue du Tensor Voting. *A condition de paramétrer correctement l'étape de filtrage qui précède, le résultat de la segmentation offre un partitionnement cohérent des différents éléments mobiles au sein du plan focal. Celui-ci est toutefois insuffisant pour construire le modèle minimal étendu, tel que proposé au chapitre 2, et la suite du manuscrit présente donc une méthode originale pour discriminer l'espace navigable et les obstacles mobiles de la scène, à partir du champ de déplacement apparent estimé ainsi que de l'image des discontinuités de ce dernier. Ainsi, le chapitre 5 propose une solution robuste de classification des points de l'image, pour déterminer s'ils appartiennent ou non au plan du sol. On emploie pour cela un modèle homographique, détaillé dans une première section, afin de définir le déplacement de l'espace navigable dans l'image. La décomposition de ce modèle permet ensuite d'obtenir la représentation géométrique du sol ainsi que la transformation de la caméra dans $\mathbb{R}^3$ entre deux acquisitions. Enfin l'identification des obstacles mobiles, traitée dans le chapitre suivant, est obtenue par l'étude du déplacement parallaxe des objets dans le plan focal. Une contrainte de rigidité fondée sur la distance des obstacles statiques à la caméra, au niveau du sol, permet de différencier les objets fixes de ceux qui sont mobiles.*

Chapitre 5

Espace navigable et odométrie visuelle

L'espace navigable correspond à la zone libre pour circuler, située devant le véhicule. Il n'est donc pas délimité par les contours d'un objet sémantique de la scène, tel que *la chaussée*, mais par les discontinuités du relief de l'espace 3-D. Le déplacement apparent de cet espace, induit par l'*ego-motion* de la caméra, est corrélé à sa géométrie. En supposant cette dernière connue, on peut alors approximer son déplacement. La comparaison de ce déplacement avec l'estimation dense du flot optique permet ensuite d'évaluer l'appartenance des points de l'image à l'espace navigable. En outre, dans l'hypothèse d'un sol plan, la transformation entre deux acquisitions est parfaitement décrite par un modèle projectif de $\mathbb{R}^{3\times3}$: l'homographie plane. Aussi, et puisque la planéité de l'espace libre est localement avérée pour la grande majorité des sites urbains et péri-urbains, la suite du document assume cette hypothèse.

La première section de ce chapitre définit l'homographie induite par un plan quelconque de l'espace 3-D entre deux prises de vue distinctes. Deux approches sont ensuite proposées pour estimer les paramètres de l'homographie induite par l'espace navigable : tout d'abord une méthode itérative originale fondée sur la mise en correspondance d'une partie de l'espace libre à l'aide d'une fonction de coût photométrique ; puis une méthode robuste reposant sur la mesure et l'évaluation du flot optique présentées aux chapitres 3 et 4. Dans chacun des cas, l'estimation du modèle homographique permet d'obtenir, par décomposition, le déplacement de la caméra entre deux acquisitions. Enfin, un dernier chapitre propose d'identifier l'espace navigable à partir du mouvement résiduel, établi en chaque point comme

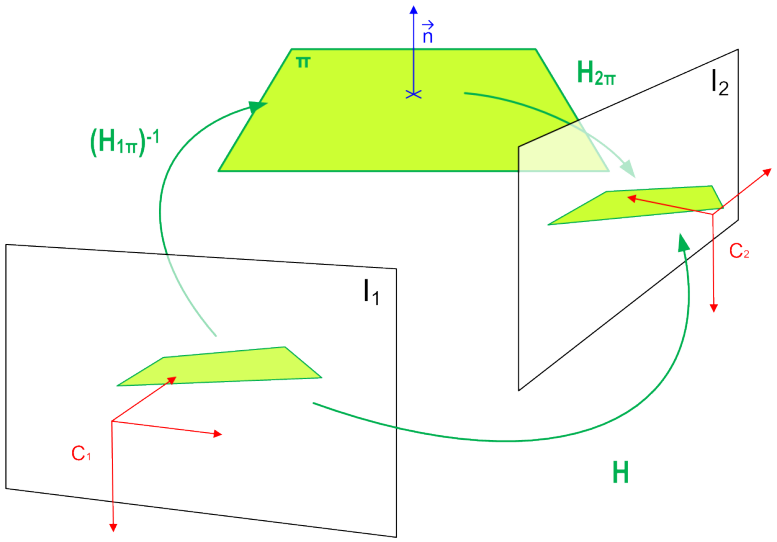


FIG. 5.1 – Homographie induite par les projections successives d'un plan de l'espace 3-D.

la distance séparant la transformation homographique et le mouvement apparent. Contrairement aux approches denses traditionnelles, qui utilisent un critère de classification photométrique, on utilise un critère dynamique, plus robuste aux erreurs du modèle.

5.1 Homographie plane

Une homographie est une application projective[1] bijective qui décrit notamment la transformation linéaire entre deux plans d'un espace 3-D [91, 4]. Dans le cadre de cette étude, on s'intéresse tout d'abord à la relation qui associe à un plan 3-D, sa projection dans une image. Soit π, un plan quelconque de $\mathbb{R}^3$, et M un point de ce plan de coordonnées $\mathbf{X} = (X,Y,Z)^T$, exprimées dans le référentiel lié à une première caméra. On note alors $\mathtt{H}_{1\pi}$, l'homographie (de $\mathbb{R}^{3\times3}$) correspondant à la projection qui transforme M en son image de coordonnées homogènes $\widetilde{\mathbf{x}}_1$:

$$\widetilde{\mathbf{x}}_1 \propto \mathtt{H}_{1\pi}\mathbf{X},\ \forall \mathbf{X} \in \pi$$

$\mathtt{H}_{1\pi}$ équivaut alors à la matrice des paramètres intrinsèques du capteur photographique, $\mathtt{K}_1 \in \mathbb{R}^{3\times3}$. De manière analogue, il existe une application projective notée $\mathtt{H}_{2\pi}$ exprimant la relation M et son image de coordonnées $\widetilde{\mathbf{x}}_2$, dans le plan focal

[1] Qui préserve la structure projective (droites, plans, etc.).

d'une seconde caméra. On peut alors établir, par composition, la transformation homographique $\mathtt{H}$, liant $\widetilde{\mathbf{x}}_1$ et $\widetilde{\mathbf{x}}_2$(Fig. 5.1) :

$$\widetilde{\mathbf{x}}_2 \propto \underbrace{\mathtt{H}_{2\pi}\mathtt{H}_{1\pi}^{-1}}_{\mathtt{H}}\widetilde{\mathbf{x}}_1 \tag{5.1}$$

On appelle $\mathtt{H}$ *l'homographie induite par le plan* π entre chaque prise de vue, et l'on considère le changement de référentiel $\mathscr{T}_{21} = [\mathbf{R} \mid \mathbf{t}]$, du repère lié à la caméra 1 vers celui associé à la caméra 2, avec $\mathbf{R} \in \mathbb{R}^{3\times3}$ la matrice de rotation et $\mathbf{t} \in \mathbb{R}^3$ le vecteur de translation. La transformation homographique $\mathtt{H}_{2\pi}$ peut alors être développée de la manière suivante :

$$\mathbf{x}_2 \propto \mathtt{H}_{2\pi}\mathbf{X} = \mathtt{K}_2\left(R\mathbf{X} + \mathbf{t}\right) \tag{5.2}$$

Soit $\mathbf{n} = (n_x, n_y, n_z)^T$ le vecteur unitaire du plan π et d la distance du plan au centre optique de la première caméra. On peut écrire :

$$\mathbf{n}^T\mathbf{X} = n_xX + n_yY + n_zZ = d,\ \forall\mathbf{X} \in \pi$$

ou encore :

$$\frac{1}{d}\mathbf{n}^T\mathbf{X} = 1,\ \forall\mathbf{X} \in \pi$$

L'équation (5.2) peut alors être formulée de la manière suivante :

$$\mathbf{x}_2 \propto \mathtt{K}_2\left(R\mathbf{X} + \mathbf{t}\frac{1}{d}\mathbf{n}^T\mathbf{X}\right) = \underbrace{\mathtt{K}_2\left(R + \mathbf{t}\frac{1}{d}\mathbf{n}^T\right)}_{\mathtt{H}_{2\pi}}\mathbf{X}$$

Finalement, l'homographie induite par le plan π, entre les prises de vue des caméras 1 et 2, s'écrit donc :

$$\mathtt{H} = \mathtt{K}_2\left(R + \mathbf{t}\frac{1}{d}\mathbf{n}^T\right)\mathtt{K}_1^{-1} \tag{5.3}$$

Les cas dégénérés, lorsque $d = 0$, ne sont pas considérés dans cette étude puisque, dans la pratique, le centre optique de la caméra ne peut appartenir au plan de l'espace navigable.

5.2 Estimation du modèle homographique

5.2.1 Méthode 1 : *patch-tracking*

L'homographie recherchée doit décrire la transformation des points de l'espace navigable, entre deux acquisitions consécutives issues d'une unique caméra. Les matrices $\mathtt{K}_1$ et $\mathtt{K}_2$ de l'équation (5.3) sont donc identiques. Le modèle homographique ne dépend alors plus que de la pose relative de la caméra par rapport au sol, du déplacement $\mathscr{T}(\mathbf{R},\mathbf{t})$ et des paramètres internes $\mathtt{K}$ de la caméra. Ces derniers sont invariants dans le temps à condition qu'aucune modification ne soit apportée à l'optique du capteur photographique. Ils sont donc calculés une seule fois, au cours d'une étape de calibration hors ligne. On emploie à cet effet la solution proposée par Zhang [86] et son implantation disponible au sein de la *Camera Calibration Toolbox* de Matlab [84].

Les paramètres d et $\mathbf{n}$ de l'équation (5.3) correspondent, respectivement, à la distance séparant le sol du centre optique de la caméra, lors de la première acquisition, et au vecteur normal de l'espace libre, exprimé dans le repère caméra au même instant. Ils sont initialisés grâce à la calibration des paramètres extrinsèques de la caméra, effectuée à l'aide de l'algorithme itératif POSIT, publié par Dementhon [88] et adapté aux points coplanaires dans [87]. Ce dernier utilise en entrée les coordonnées 3-D d'un amer de dimensions connues, contenu dans le plan de l'espace navigable, tel que le dessin d'un passage piéton (Fig. 5.2). On considère par ailleurs que la distance d est invariante au cours du temps, malgré les effets de roulis et de tangage du véhicule. Ceci est motivé par la position centrale de la caméra, derrière le pare-brise : lorsqu'une partie de la voiture s'abaisse lors d'un freinage, d'une accélération ou d'un changement de direction, la partie opposée se surélève de sorte qu'au centre du véhicule, la variation de la distance au sol est négligeable.

Connaissant la position relative de la caméra par rapport au sol ainsi que la focale du capteur et le centre de l'image, on projette dans le plan rétinien une grille 3-D, virtuellement posée sur la voie (Fig. 5.3). L'objectif consiste ensuite à trouver la transformation $\mathscr{T}(\mathbf{R},\mathbf{t})$ qui, injectée dans l'équation (5.3), permet alors de calculer les coordonnées de cette grille dans l'image courante. Le problème revient donc à minimiser une fonction coût photométrique $f(\mathbf{R},\mathbf{t})$, entre les points de cette grille dans l'image précédente, transformés selon l'homographie estimée, et ceux de l'image courante. Pour satisfaire la contrainte temps-réel, on choisit de modéliser f par la somme des différences absolues sur l'intensité de ces points,

FIG. 5.2 – Sélection des points d'un amer contenu dans le plan de l'espace navigable, et dont les coordonnées 3-D sont connues, pour la calibration des paramètres extrinsèques de la caméra.

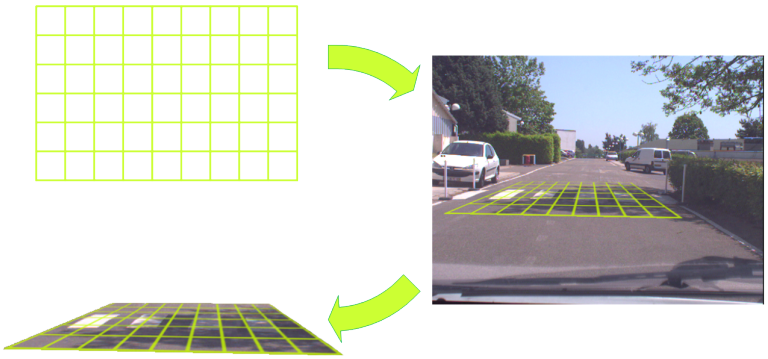


FIG. 5.3 – Projection d'un plan 3-D dans l'image.

pour chacune des composantes de la couleur, R, G et B. Soit $\mathtt{H}(\mathbf{R},\mathbf{t})$, l'homographie induite par le plan de l'espace navigable associée au déplacement $\mathscr{T}(\mathbf{R},\mathbf{t})$ de la caméra entre deux acquisitions. On note Ω, l'ensemble des pixels, de coordonnées $\mathbf{x} = (x,y)$, de la grille projetée dans la première image I_1. La fonction de coût f s'écrit alors :

$$f(\mathbf{R},\mathbf{t}) = \frac{1}{3\|\Omega\|}\sum_{\mathbf{x}}^{\Omega}\sum_{c}^{rgb} |I_1(\mathbf{x},c) - I_2(\mathtt{H}(\mathbf{R},\mathbf{t})\mathbf{x},c)|$$

avec $I(\mathbf{x},c)$ l'intensité de la composante couleur c, pour point de coordonnées $\mathbf{x}$ dans l'image I.

A l'aide du logiciel Ggobi, conçu pour la visualisation des données de grande dimension, une étude exhaustive de la fonction coût, dans l'espace des transforma-

tions, est réalisée autour de la solution optimale. Elle met en avant l'absence de minima local à proximité de cette dernière. En outre, la fréquence d'acquisition de la caméra est supposée suffisamment élevée pour considérer la vitesse du véhicule quasi constante sur trois acquisitions consécutives. Le processus de minimisation est alors initialisé par le déplacement estimé sur la paire d'images précédente. Une simple descente de gradient suffit ensuite pour converger vers la solution optimale. En notant $\mathscr{T}_i = (Rx_i, Ry_i, Rz_i, tx_i, ty_i, tz_i)$ les paramètres du déplacement calculés à l'itération i, la solution à l'itération suivante est donnée par :

$$\mathscr{T}_{i+1} = \mathscr{T}_i + \triangle\mathscr{T}$$

avec :

$$\triangle\mathscr{T} = -\eta\nabla f(\mathscr{T})$$

où ∇f désigne le gradient de la fonction coût et η le pas d'entraînement du processus itératif de minimisation. η est choisi de manière adaptative en fonction de la pente calculée afin de converger rapidement vers la solution optimale. De son côté, ∇f peut être exprimé comme la somme des dérivées partielles de f en fonction des paramètres du déplacement :

$$\nabla f(\mathscr{T}) = \frac{\partial f}{\partial Rx}(\mathscr{T}) + \frac{\partial f}{\partial Ry}(\mathscr{T}) + \cdots + \frac{\partial f}{\partial tz}(\mathscr{T})$$

La grille est suivie au fil des acquisitions successives jusqu'à ce qu'elle ne soit plus intégralement contenue dans l'image. Une nouvelle grille est alors projetée. La normale au sol $\mathbf{n}$, exprimée dans le repère caméra lié à la première prise de vue, pour chaque paire d'images considérée, est mise à jour en fonction des paramètres de déplacement $\mathscr{T}(\mathbf{R}, \mathbf{t})$ précédemment calculés.

L'intégration est réalisée sur une architecture mono-core, équipée d'un processeur de type Pentium III cadencé à 3Ghz. L'approche est alors évaluée à l'aide d'un GPS centimétrique RTK de manière à comparer la pose relative obtenue avec la localisation du système de positionnement par satellite (Fig. 5.4). On constate une dérive, inhérente à toute méthode de positionnement relatif, inférieure à 1% sur les cent premiers mètres de l'exemple présenté. Cette dérive augmente au-delà de 130 mètres, du fait d'un changement de pente à cet endroit de la scène ainsi que d'une trop grande uniformité de la texture au sol sur la portion de route qui suit. L'estimation du modèle homographique, vue comme un problème de recherche opérationnelle dans l'espace des transformations de $\mathbb{R}^3$, permet donc de suppléer

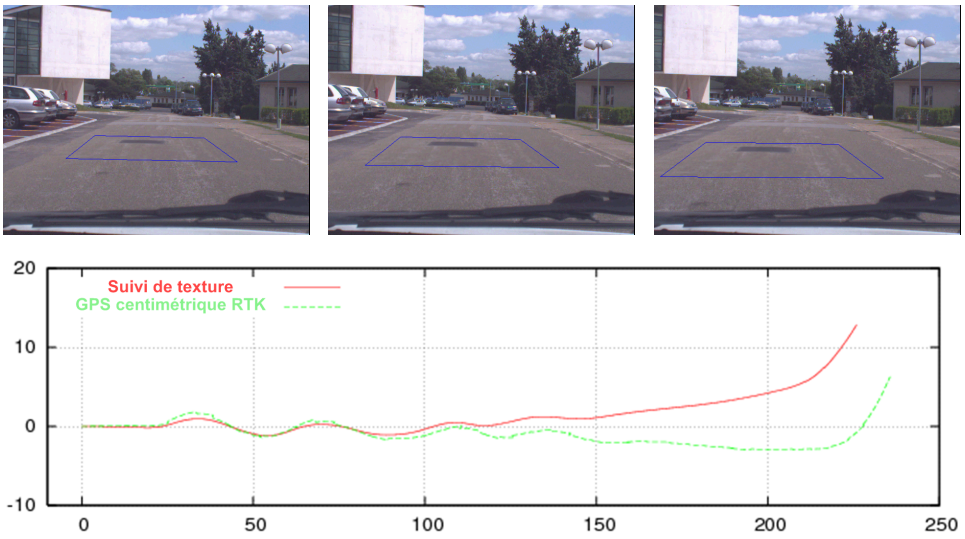


FIG. 5.4 – Evaluation de l'odométrie visuelle par suivi de texture (en rouge sur le graphique) et à l'aide d'un GPS RTK (en vert), pour la séquence *Route*.

temporairement un capteur GPS classique afin de pallier aux éventuelles ruptures du signal satellitaire. Elle permet également de fournir l'homographie induite par le plan associé à l'espace navigable. Toutefois, cette approche ne tire pas partie de la mise en correspondance dense, réalisée avec l'estimation du flot optique. Aussi, la corrélation des points de la grille paraît-elle redondante avec les opérations précédentes. Enfin, de même que l'on constate une dérive sur la position relative mesurée, la normale au plan du sol est progressivement faussée par l'accumulation des erreurs lors des mises à jour successives. Par conséquent, l'homographie, calculée à l'aide de la formule (5.3), diverge de plus en plus de la solution optimale.

5.2.2 Méthode 2 : estimation directe et décomposition

Une seconde méthode consiste à estimer l'homographie recherchée, directement à l'aide des paires de points mis en correspondance lors de l'étape de calcul du flot optique [100, 93], avant de la décomposer pour obtenir le déplacement du capteur.

La matrice homographique $\mathtt{H} \in \mathbb{R}^{3\times3}$ se compose de 9 entrées dont 1 facteur d'échelle, elle possède ainsi 8 degrés de liberté. Pour toute paire de points de coordonnées respectives $\mathbf{x}$ et $\mathbf{x}'$, tels que $\mathbf{x}' \propto \mathtt{H}\mathbf{x}$, chacun des deux degrés de liberté associés à $\mathbf{x}'$ doit satisfaire la transformation homographique de $\mathbf{x}$. Le nombre de couples de points 2-D, nécessaires au calcul d'une homographie, est donc égal à 4.

Soit $\mathbf{h}_i$, i$\in \{1;2;3\}$ les vecteurs lignes de H, la transformation homographique de $\mathbf{x}$ peut alors s'écrire :

$$\mathtt{H}\mathbf{x} = \begin{bmatrix} \mathbf{h}_1\mathbf{x} \\ \mathbf{h}_2\mathbf{x} \\ \mathbf{h}_3\mathbf{x} \end{bmatrix}$$

Puisque ces vecteurs sont colinéaires, le produit vectoriel des vecteurs homogènes $\mathbf{x}'$ et H$\mathbf{x}$ conduit à une forme linéaire simple, $\mathbf{x}' \times \mathtt{H}\mathbf{x} = \mathbf{0}$, dont il est facile de déduire H. Le développement de cette expression permet d'obtenir :

$$\mathbf{x}' \times \mathtt{H}\mathbf{x} = \begin{bmatrix} y'\mathbf{h}_3\mathbf{x} - z'\mathbf{h}_2\mathbf{x} \\ z'\mathbf{h}_3\mathbf{x} - x'\mathbf{h}_2\mathbf{x} \\ x'\mathbf{h}_3\mathbf{x} - y'\mathbf{h}_2\mathbf{x} \end{bmatrix}$$

et le système linéaire suivant :

$$\begin{bmatrix} 0 & -z'\mathbf{x}^T & y'\mathbf{x}^T \\ z'\mathbf{x}^T & 0 & -x'\mathbf{x}^T \\ -y'\mathbf{x}^T & x'\mathbf{x}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{bmatrix} = \mathbf{0} \tag{5.4}$$

contenant seulement deux équations linéairement indépendantes. On conserve les deux premières lignes du système (5.4), qui s'écrivent alors :

$$\underbrace{\begin{bmatrix} 0 & -z'\mathbf{x}^T & y'\mathbf{x}^T \\ z'\mathbf{x}^T & 0 & -x'\mathbf{x}^T \end{bmatrix}}_{\mathbf{A}\in\mathbb{R}^{2\times 9}} \underbrace{\begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{bmatrix}}_{\mathbf{h}\in\mathbb{R}^9} = \mathbf{0} \tag{5.5}$$

Calculer l'homographie H revient donc à résoudre l'expression $\mathbf{Ah} = \mathbf{0}$ de manière à trouver les solutions non triviales de $\mathbf{h}$. Toutefois, si $\mathbf{h}$ est solution du système (5.5), pour tout scalaire k, $k\mathbf{h}$ est également solution de ce dernier. On fixe donc une contrainte sur la norme du vecteur $\mathbf{h}$ que l'on impose unitaire, de manière à s'assurer d'une solution non nulle. En l'absence de solution exacte due erreurs de mesure, le problème revient à minimiser $\|\mathbf{Ah}\|$.

Or, pour toute matrice $\mathbf{M} \in \mathbb{R}^{m\times n}$, et donc pour $\mathbf{A}$, il existe une décomposition en valeurs singulières (SVD[2] [101]) pour laquelle $\mathbf{M} = \mathbf{UDV}^T$, avec :

– $\mathbf{U} \in \mathbb{R}^{m\times n}$, une matrice dont les n vecteurs colonnes sont orthogonaux,

[2] ***S**ingular **V**alue **D**ecompostion.*

- $\mathbf{D} \in \mathbb{R}^{n\times n}$, une matrice diagonale non négative, ordonnée de sorte que $D_{i,i} \geq D_{i+1,i+1}, i \in [1; n-1]$,
- $\mathbf{V}^T \in \mathbb{R}^{n\times n}$, une matrice orthogonale.

Et puisque la multiplication d'un vecteur par une matrice orthogonale préserve la norme de ce vecteur, il est possible d'écrire :

$$\left\|\mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{h}\right\| = \left\|\mathbf{D}\mathbf{V}^T\mathbf{h}\right\|$$

et :

$$\left\|\mathbf{V}^T\mathbf{h}\right\| = \|\mathbf{h}\|$$

En posant $\mathbf{y} = \mathbf{V}^T\mathbf{h}$, minimiser $\|\mathbf{A}\mathbf{h}\|$ avec $\|\mathbf{h}\| = 1$ revient donc à minimiser $\|\mathbf{D}\mathbf{y}\|$, avec $\|\mathbf{y}\| = 1$. La solution correspond alors au vecteur $\mathbf{y} = (0, \cdots, 0, 1)^T$. Enfin, comme $\mathbf{h} = \mathbf{V}^T\mathbf{y}$, le vecteur $\mathbf{h}$ correspond finalement à la dernière colonne de la matrice orthogonale $\mathbf{V}$, et :

$$\mathtt{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_6 & h_8 & h_9 \end{bmatrix}$$

Il est donc possible de calculer directement l'homographie induite par le plan du sol, à partir de la mise en correspondance de quatre de ses points, dans deux prises de vue distinctes. La difficulté revient à sélectionner quatre couples de points appartenant à l'image de l'espace navigable. Pour cela, on suppose la superficie de cet espace, dans le plan focal, supérieure à celle de tout autre plan de la scène. Une méthode robuste, décrite dans la prochaine section, permet alors d'exclure les *outliers* du processus d'estimation de l'homographie. On appelle *outlier* tout couple de points mal appariés ou n'appartenant pas au modèle recherché.

Estimation robuste par consensus (RANSAC)

Fishler et Bolles publient en 1981, une méthode itérative pour déterminer les paramètres d'un modèle mathématique à partir d'un ensemble de données observées, tout en jugeant pour chacune d'elle si sa mesure est cohérente (*inlier*) ou non (*outlier*). RANSAC[3] est un algorithme non déterministe, pour lequel la solution est établie avec une certaine probabilité. Cette dernière croît avec le nombre d'itérations effectuées.

[3] ***RAN****dom* ***SA****mple* ***C****onsensus.*

Le principe de l'algorithme consiste à sélectionner aléatoirement, parmi l'ensemble des données de départ, noté $\mathscr{S}$, un échantillon de taille s, suffisante pour calculer le modèle recherché. On appelle *support* de ce modèle, le nombre d'éléments considérés comme *inliers*, c'est-à-dire dont la distance au modèle est inférieure à un seuil prédéfini. Le processus est répété jusqu'à assurer l'obtention d'un résultat correct avec une probabilité p. Cette dernière équivaut à la probabilité de tirer s *inliers* au sein d'un même échantillon, au cours des K itérations du processus.

On note w^s, la probabilité de tirer s *inliers*, $(1-w^s)$ désigne donc la probabilité qu'au moins 1 *outlier* soit présent dans chaque échantillon, aussi :

$$1-p=(1-w^s)^K$$

et :

$$K=\frac{\log(1-p)}{\log(1-w^s)}$$

On pose généralement $p=0.99$. K ne dépend plus alors que de la proportion $\varepsilon=1-w$ d'*outliers* présents dans l'ensemble $\mathscr{S}$ de départ. La recherche d'une homographie ($s=4$) nécessite ainsi 72 itérations lorsque $\varepsilon=50\%$, contre seulement 5 si $\varepsilon=10\%$.

Cependant, la proportion d'*inliers* contenus dans $\mathscr{S}$ n'est généralement pas connue. Dans ce cas, K est mis à jour à la fin de chaque itération, en fonction du nombre d'*outliers* correspondant au modèle de support maximal δ_{max} :

$$K=\frac{\log(1-p)}{\log(1-(\delta_{max}/|\mathscr{S}|)^s)}$$

La fonction distance, servant à déterminer le support du modèle homographique courant, mesure l'*erreur de transfert symétrique*, notée ξ, associée à toute paire de points $\{\mathbf{x},\mathbf{x}'\}$. Elle correspond à la somme des erreurs de projection et de re-projection, liées à la transformation $\mathtt{H}$ considérée, et s'écrit donc :

$$\xi\left(\mathbf{x},\mathbf{x}'\right)=d\left(\mathbf{x},\mathtt{H}^{-1}\mathbf{x}'\right)^2+d\left(\mathbf{x}',\mathtt{H}\mathbf{x}\right)^2 \tag{5.6}$$

où $d(\mathbf{x},\mathbf{y})$ désigne la distance euclidienne séparant $\mathbf{x}$ et $\mathbf{y}$.

L'ensemble $\mathscr{S}$, des couples de points appariés, est établi à l'aide d'une grille de discrétisation appliquée sur l'image. Chaque cellule de la grille étant représentée par le point dont la mise en correspondance, dans la seconde image, correspond

Algorithm 3 RANdom SAmple Consensus

ENTRÉE(S): Ensemble S des points mis en correspondance.
ENTRÉE(S): Probabilité p que 4 éléments $s_i \in S$ satisfassent H.
ENTRÉE(S): Ecart type σ sur l'erreur de mesure des éléments de $\mathscr{S}$.
SORTIE(S): Modèle homographique H prédominant.

1: **POUR** K échantillons de 4 points sélectionnés au hasard parmi $\mathscr{S}$ **FAIRE**
2: Estimer le modèle homographique H.
3: Calculer la distance $d_\perp$ au modèle pour chaque paire de points.
4: Déterminer le nombre N d'*inliers* sur $\mathscr{S}$ tels que $d_\perp < T$.
5: Fixer la probabilité ε qu'un élément de $\mathscr{S}$ soit un *outlier*.
6: Mettre à jour le nombre d'itérations nécessaires pour sélectionner un jeu de 4 *inliers*.
7: Retourner H pour lequel N est maximal.

au maximum de vraisemblance. De cette façon, on limite le nombre de calculs effectués lors de la mesure du support, à chaque itération de l'algorithme RANSAC. La sélection du meilleur candidat est réalisée en fonction du score obtenu à l'issue de l'étape de *Tensor Voting*, en accord avec le résultat illustré Fig. 4.8. $\mathscr{S}$ est également limité aux points contenus dans le polygone convexe associé à l'espace navigable identifié grâce à la paire d'images précédente.

On ajoute par ailleurs une contrainte liée à la cohérence spatiale du modèle homographique. Ce dernier étant induit par un plan dont la projection n'est pas définie dans toute l'image et sur lequel on circule, il n'a de sens, qu'en deça de la ligne d'horizon. Dans le cas de la transformation image-à-image de l'espace navigable, l'expression de l'homographie correspondante doit donc être limitée à la région située en deçà de la ligne d'horizon. L'ordonnée de cette dernière, au point 2-D d'abscisse $u_l = u_0$, est donnée par la relation suivante :

$$v_l = v_0 - f \sin(\alpha) = v_0 - f\frac{n_z}{n_y}$$

avec α, l'inclinaison de la caméra autour de son axe transversal et f, la distance focale de la caméra (Fig. 5.5). L'inclinaison de la ligne d'horizon dans le plan image est donnée par :

$$\beta = \arcsin\left(\frac{n_x}{n_y}\right)$$

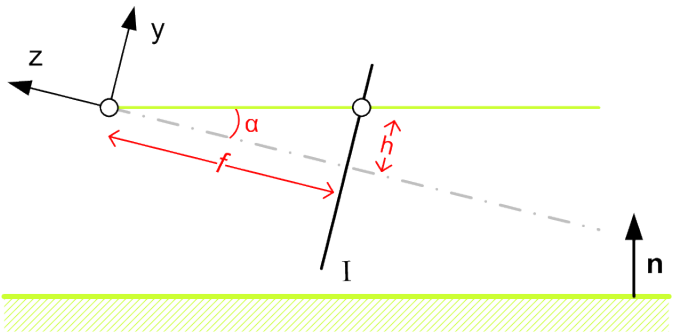


FIG. 5.5 – Projection de la ligne d'horizon correspondant au plan de l'espace navigable, pour le point d'abscisse $u_l = u_0$.

Décomposition de la matrice homographique

Les paramètres $\left\{R, \frac{\mathbf{t}}{d}, \mathbf{n}\right\}$, décrivant le déplacement de la caméra entre deux acquisitions, peuvent être obtenus par décomposition de la matrice homographique $\mathtt{H}_c$, définie par :

$$\mathtt{H}_c = \mathtt{K}^{-1}\mathtt{H}\mathtt{K} = R + \mathbf{t}\frac{1}{d}\mathbf{n}^T$$

Cette opération est couramment réalisée à l'aide de méthodes numériques telles que celles proposées par Faugeras [90] ou Zhang [92]. Elles reposent sur la décomposition en valeurs propres de la matrice homographique et ne permettent donc pas d'obtenir l'expression des paramètres $\left\{R, \frac{\mathbf{t}}{d}, \mathbf{n}\right\}$ en fonction des composantes de $\mathtt{H}_c$. Bien que très largement utilisées dans la littérature, on leur préfère, dans le cadre de cette étude, la solution analytique présentée par Malis et Vargas [94], dont l'implantation n'est pas sujette aux erreurs numériques. Quatre jeux de paramètres sont ainsi déduits de $\mathtt{H}_c$, répartis en deux solutions totalement distinctes et leurs opposés :

$$\begin{aligned} Rtn_a &= \{R_a, \mathbf{t}_a, \mathbf{n}_a\} \\ Rtn_b &= \{R_b, \mathbf{t}_b, \mathbf{n}_b\} \\ Rtn_{a-} &= \{R_a, -\mathbf{t}_a, -\mathbf{n}_a\} \\ Rtn_{b-} &= \{R_b, -\mathbf{t}_b, -\mathbf{n}_b\} \end{aligned} \tag{5.7}$$

La suite du document ne présente que les principales étapes du calcul de cette décomposition. Les détails du développement sont laissés à la discrétion du lecteur qui pourra se référer à [94].

On introduit tout d'abord la matrice symétrique $\mathbf{S}$, définie à partir de la matrice

homographique $\mathtt{H}_c$, de la manière suivante :

$$\mathbf{S} = \mathtt{H}_c^T \mathtt{H}_c - \mathtt{I} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}$$

L'opposé des mineurs de la matrice $\mathbf{S}$, notés $\mathbf{M}_{Sij}, i, j \in \{1;2;3\}$, correspond à l'opposé des déterminants des sous-matrices de $\mathbf{S}$. Ainsi, par exemple :

$$\mathbf{M}_{S11} = -\begin{vmatrix} s_{22} & s_{23} \\ s_{23} & s_{33} \end{vmatrix} = s_{23}^2 - s_{22}s_{33} \geq 0$$

Le vecteur normal $\mathbf{n}_k, k \in \{a;b\}$, à partir duquel seront par la suite calculés les paramètres associés R_k et $\mathbf{t}_k$, peut alors être obtenu de trois manières différentes :

$$\mathbf{n}_k(s_{ii}) = \frac{\mathbf{n}'_k(s_{ii})}{\left\|\mathbf{n}'_k(s_{ii})\right\|}$$

avec :

$$\mathbf{n}'_a(s_{11}) = \begin{bmatrix} s_{11} \\ s_{12} + \sqrt{\mathbf{M}_{S33}} \\ s_{13} + \varepsilon_{23}\sqrt{\mathbf{M}_{S22}} \end{bmatrix}; \quad \mathbf{n}'_b(s_{11}) = \begin{bmatrix} s_{11} \\ s_{12} - \sqrt{\mathbf{M}_{S33}} \\ s_{13} - \varepsilon_{23}\sqrt{\mathbf{M}_{S22}} \end{bmatrix}$$

$$\mathbf{n}'_a(s_{22}) = \begin{bmatrix} s_{12} + \sqrt{\mathbf{M}_{S33}} \\ s_{22} \\ s_{23} - \varepsilon_{13}\sqrt{\mathbf{M}_{S11}} \end{bmatrix}; \quad \mathbf{n}'_b(s_{22}) = \begin{bmatrix} s_{12} - \sqrt{\mathbf{M}_{S33}} \\ s_{22} \\ s_{23} + \varepsilon_{13}\sqrt{\mathbf{M}_{S11}} \end{bmatrix}$$

$$\mathbf{n}'_a(s_{33}) = \begin{bmatrix} s_{13} + \varepsilon_{12}\sqrt{\mathbf{M}_{S22}} \\ s_{23} + \sqrt{\mathbf{M}_{S11}} \\ s_{33} \end{bmatrix}; \quad \mathbf{n}'_b(s_{33}) = \begin{bmatrix} s_{13} + \varepsilon_{12}\sqrt{\mathbf{M}_{S22}} \\ s_{23} + \sqrt{\mathbf{M}_{S11}} \\ s_{33} \end{bmatrix}$$

et $\varepsilon_{ij} = 1$ lorsque $\mathbf{M}_{Sij} \geq 0$, -1 sinon. L'unique cas singulier correspond à l'homographie induite par une rotation pure, avec donc $\mathtt{H}_c = R$. Chaque entrée de la matrice S est alors nulle. Une telle hypothèse n'est cependant pas envisageable dans le cadre de cette étude, les modèles de déplacement des véhicules excluant cette possibilité.

L'expression du vecteur de translation dans le repère de référence, $\mathbf{t}_k^* = R_k^T \mathbf{t}_k$,

peut être obtenue à l'aide du vecteur normal $\mathbf{n}_k$ précédemment calculé, de la manière suivante :

$$\mathbf{t}_k^*(s_{11}) = \frac{\left\|\mathbf{n}_k'(s_{11})\right\|}{2s_{11}} \begin{bmatrix} s_{11} \\ s_{12} \mp \sqrt{\mathbf{M}_{S33}} \\ s_{13} \mp \varepsilon_{23}\sqrt{\mathbf{M}_{S22}} \end{bmatrix} - \frac{\|\mathbf{t}_k\|^2}{2\left\|\mathbf{n}_k'(s_{11})\right\|} \begin{bmatrix} s_{11} \\ s_{12} \pm \sqrt{\mathbf{M}_{S33}} \\ s_{13} \pm \varepsilon_{23}\sqrt{\mathbf{M}_{S22}} \end{bmatrix}$$

$$\mathbf{t}_k^*(s_{22}) = \frac{\left\|\mathbf{n}_k'(s_{22})\right\|}{2s_{22}} \begin{bmatrix} s_{12} \mp \sqrt{\mathbf{M}_{S33}} \\ s_{22} \\ s_{23} \mp \varepsilon_{13}\sqrt{\mathbf{M}_{S11}} \end{bmatrix} - \frac{\|\mathbf{t}_k\|^2}{2\left\|\mathbf{n}_k'(s_{22})\right\|} \begin{bmatrix} s_{12} \pm \sqrt{\mathbf{M}_{S33}} \\ s_{22} \\ s_{23} \pm \varepsilon_{13}\sqrt{\mathbf{M}_{S11}} \end{bmatrix}$$

$$\mathbf{t}_k^*(s_{33}) = \frac{\left\|\mathbf{n}_k'(s_{33})\right\|}{2s_{33}} \begin{bmatrix} s_{13} \mp \varepsilon_{12}\sqrt{\mathbf{M}_{S22}} \\ s_{23} \mp \sqrt{\mathbf{M}_{S11}} \\ s_{33} \end{bmatrix} - \frac{\|\mathbf{t}_k\|^2}{2\left\|\mathbf{n}_k'(s_{33})\right\|} \begin{bmatrix} s_{13} \pm \varepsilon_{12}\sqrt{\mathbf{M}_{S22}} \\ s_{23} \pm \sqrt{\mathbf{M}_{S11}} \\ s_{33} \end{bmatrix}$$

en considèrant alors l'opérateur supérieur décrit par les symboles $\pm$ et $\mp$, lorsque $k = a$, et l'opérateur inférieur si $k = b$. De même, on définit :

$$\|\mathbf{t}_e\|^2 = 2 + \text{trace}(\mathbf{S}) - v$$

avec :

$$v = 2\sqrt{1 + \text{trace}(\mathbf{S}) - \mathbf{M}_{S11} - \mathbf{M}_{S22} - \mathbf{M}_{S33}}$$

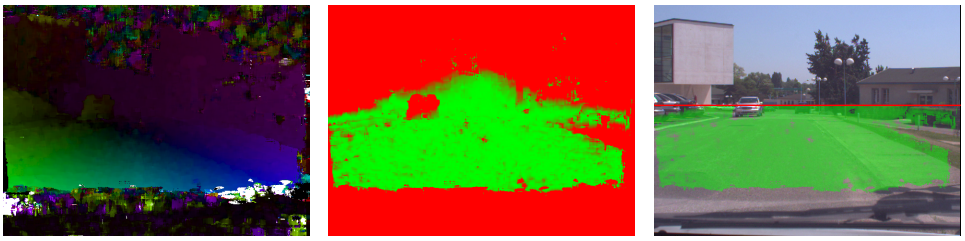
La décomposition finale est identique quelle que soit l'expression utilisée. Toutefois, le calcul de la translation faisant intervenir une division par s_{ii}, on utilise généralement la solution associée à la composante s_{ii} de valeur absolue maximale. Enfin, la matrice de rotation est donnée par :

$$R_k = \mathtt{H}\left(\mathtt{I} - \frac{2}{v}\mathbf{t}_k^*\mathbf{n}_k^T\right)$$

et :

$$\mathbf{t}_k = R_k\mathbf{t}_k^*$$

L'expression de la translation est donnée à un facteur d'échelle près, avec $\mathbf{t}_k = \mathbf{t}/d$. Il est donc nécessaire de connaître la distance séparant le sol du centre optique de la caméra. Tout comme dans l'approche par suivi de texture, on suppose que cette distance est invariante dans le temps. Une simple mesure hors ligne de celle-ci est



(a) Flot optique. (b) Erreur de transfert symétrique. (c) Ligne d'horizon.

FIG. 5.6 – Identification de l'espace navigable, à partir de l'erreur de transfert symétrique (Eq. 5.6), entre le flot optique et l'homographie induite par le plan de l'espace navigable. Du vert au rouge, dans le sens des valeurs croissantes de l'erreur.

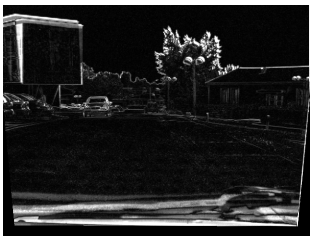
donc suffisante.

Afin de réduire le nombre de solutions possibles, on applique tout d'abord la *contrainte de profondeur positive* : puisque la caméra ne peut observer les points de la face cachée d'un plan, la composante n_z du vecteur normal doit être positive. De cette manière, deux solutions sur les quatres présentées en (5.7) sont éliminées. Le choix parmi les deux ensembles de paramètres restants est effectué en bornant l'amplitude maximale de tangage et de roulis du véhicule. Dans la pratique, ces deux solutions sont effectivement suffisamment éloignées, pour ne pas être confondues Enfin, la pose de la caméra peut être initialisée au cours d'une étape, hors ligne, de calibration des paramètres extrinsèques, en même temps que la mesure de d, nécessaire au calcul du vecteur de translation.

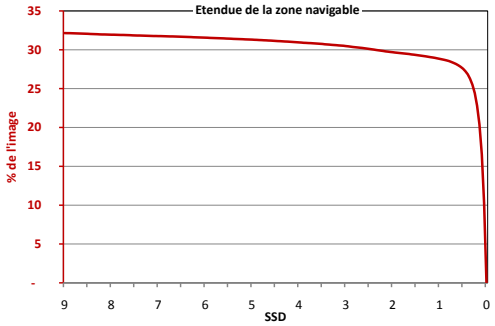
Segmentation de l'espace navigable

L'identification de l'espace navigable à partir de son homographie équivaut à déterminer le support de cette transformation, au sein du champ de déplacement issu de la mesure du flot optique. Il est ainsi possible d'obtenir une représentation dense, non seulement de l'espace libre pour circuler, mais également, de manière duale, des obstacles tri-dimensionnels adjacents (Fig. 5.6). Chaque point de l'image est évalué sur un critère d'appartenance à l'une de ces deux classes, en fonction de son mouvement apparent. A l'inverse, les approches fondées sur l'estimation du modèle homographique, à partir d'une mise en correspondance éparse, ne peuvent qu'identifier les dissemblances entre l'image courante, transformée par ce modèle, et l'image suivante. La figure 5.7 compare ainsi, pour une même homo-

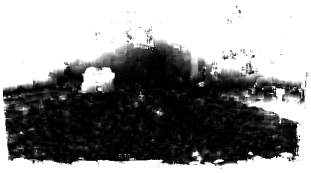
(a) Séquence originale.



(b) Erreur sur la luminance.



(c) Etendue de l'espace navigable après seuillage.



(d) Erreur sur le flot optique.

FIG. 5.7 – Erreur de transfert symétrique établie à partir de l'homographie du plan du sol, sur la luminance des points de l'image (b) et sur la mesure du flot optique (d) (du noir au blanc, dans le sens des valeurs croissantes de l'erreur). L'étendue de l'espace navigable en fonction du seuil appliqué sur (d) est donnée graphique (c). La segmentation de l'espace navigable y apparaît robuste au seuillage.

graphie, l'erreur de transfert symétrique appliquée sur le flot optique (Fig. 5.7(d)) et sur la luminance en chaque point de l'image (Fig. 5.7(b)). Dans ce dernier cas, seules les discontinuités photométriques permettent de distinguer les objets du sol. Elles sont uniquement localisées aux contours des régions de texture uniforme et difficilement discernables des discontinuités induites par l'approximation du modèle homographique, au niveau des forts gradients. Les approches photométriques garantissent donc, au mieux, l'identification des obstacles dans la scène.

Enfin, on intègre temporellement le résultat de la fonction distance, de façon à lisser les erreurs introduites lors l'estimation du flot optique ou par l'approximation du modèle homographique. Pour cela, on conserve, à l'aide d'une structure de type pile FIFO[4], les n dernières mesures de l'erreur géométrique (5.6) associée à chaque

[4] ***F**irst **I**n **F**irst **O**ut.*

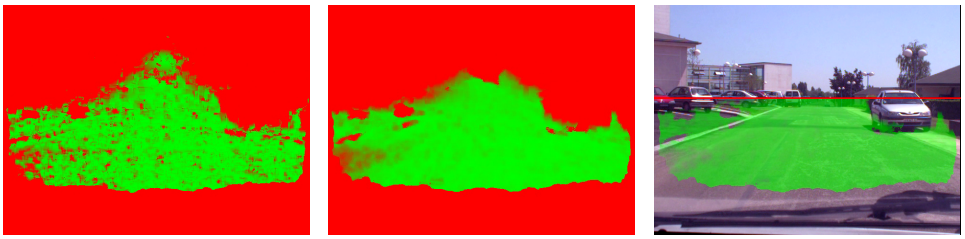


FIG. 5.8 – Représentation de l'erreur géométrique entre le modèle homographique et la mesure du flot optique, avec (au centre) et sans filtrage temporel (à gauche).

point de l'image, pondérées à l'aide d'une fonction gaussienne centrée sur la mesure courante. Comme pour l'estimation de l'homographie, le résultat du *Tensor Voting* est employé afin d'ignorer, pour chaque nouvelle acquisition, les couples de points aberrants. La pile des pixels concernés reste alors inchangée. Le résultat du processus est illustré par la figure 5.8 : l'espace libre y apparaît mieux défini et moins sensible aux bruits de mesure que dans le résultat de la version non lissée.

5.3 *Inverse Perspective Mapping* (IPM)

En fonction du contexte, modéliser l'environnement peut s'avérer nécessaire, notamment pour reproduire en conduite automatique un trajet précédemment parcouru de façon supervisée. On parle alors de SLAM[5], lorsque la modélisation est réalisée en ligne. Initialement associé à la reconstruction de cartes 2-D par scanner laser, le SLAM constitue aujourd'hui un domaine de recherche à part entière, en vision artificielle. Pour des raisons principalement techniques, les méthodes répertoriées sont non denses et reposent sur le suivi, dans $\mathbb{R}^3$, de patchs texturés. La suite du chapitre présente une approche originale permettant de construire un modèle texturé 3-D dense de l'espace navigable, à partir de deux acquisitions. Le lecteur désirant concevoir un module de SLAM pourra intégrer temporellement ce résultat, grâce à l'estimation de l'*ego-motion* issue de la décomposition homographique (section 5.2.2).

Considérant l'espace navigable plan et son modèle homographique, le calcul des coordonnées métriques de son image est immédiat. Le modèle 3-D ainsi obtenus est toutefois d'autant plus éparse que l'on s'éloigne du centre optique de la caméra. Afin de le rendre dense, il est nécessaire de définir la transformation qui, à tout point visible du sol, fait correspondre un point de l'espace projectif $\mathbb{P}^2$,

[5] ***S**imultaneous **L**ocalization **A**nd **M**apping.*

contenu dans l'image. Il existe à ce sujet, dans la littérature, plusieurs équations permettant d'annuler les effets de la perspective, en formant une vue *de dessus*, couramment appelée IPM[6][20]. Cependant, ces équations nécessitent le plus souvent de connaître l'angle d'ouverture de la caméra, ou encore de sélectionner différents points caractéristiques au sein de l'image. La définition du champ de vision final n'est jamais directement associée à la profondeur des éléments observés. On propose donc une formulation mieux adapté à la modélisation de l'espace navigable, dépendant uniquement de la taille de l'image, de la profondeur de champ souhaitée, ainsi que de la normale au plan du sol. Enfin, puisqu'il s'agit d'une isotropie de $\mathbb{P}^2$ vers $\mathbb{P}^2$, la solution présentée consiste en une homographie notée $\mathtt{H}_{ipm}$ (Fig. 5.9).

Soient $\mathbf{x}_{ipm} = (x_{ipm}, y_{ipm})^T$ les coordonnées d'un point quelconque de l'image obtenue après transformation IPM. On note w la largeur de l'image, h sa hauteur, d_b et d_f les profondeurs bornant respectivement le début et la fin du champ de vision. Les coordonnées métriques $\mathbf{X}(\mathbf{x}_{ipm}) = (X,Y,Z)$ associées à tout point de cette image sont données par le système suivant :

$$\begin{aligned} X(\mathbf{x}_{ipm}) &= \frac{1}{2}\cdot\frac{w}{h}\cdot\frac{d_f}{x_0}\cdot(x_{ipm}-x_0) \\ Z(\mathbf{x}_{ipm}) &= d_f\cdot\left(1-\frac{y_{ipm}}{h}\right)+d_b \\ Y(\mathbf{x}_{ipm}) &= \frac{d}{n_y}+\frac{n_x}{n_y}\cdot X(\mathbf{x}_{ipm})+\frac{n_z}{n_y}\cdot Z(\mathbf{x}_{ipm}) \end{aligned}$$

avec (x_0, y_0) le centre de l'image, $\mathbf{n} = (n_x, n_y, n_z)^T$ la normale au sol et d la distance sol-caméra. La forme matricielle du passage en coordonnées métrique s'écrit donc :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{2}\cdot\frac{wd_f}{hx_0} & 0 & -\frac{1}{2}\cdot\frac{wd_f}{h} \\ -\frac{1}{2}\cdot\frac{n_x w d_f}{n_y h x_0} & \frac{n_z d_f}{n_y h} & \frac{1}{2}\cdot\frac{n_x w d_f}{n_y h}+\frac{d}{n_y}-\frac{n_z}{n_y}(d_f+d_b) \\ 0 & -\frac{d_f}{h} & d_f+d_b \end{pmatrix}}_{\mathtt{A}} \begin{pmatrix} x_{ipm} \\ y_{ipm} \\ 1 \end{pmatrix}$$

Aussi la position des points de l'image courante, d'après leurs coordonnées IPM, peut-elle être formulée de la façon suivante :

$$\widetilde{\mathbf{x}} \propto \mathtt{K}\mathtt{A}\widetilde{\mathbf{x}}_{ipm}$$

[6] ***I**nverse **P**erspective **M**apping.*

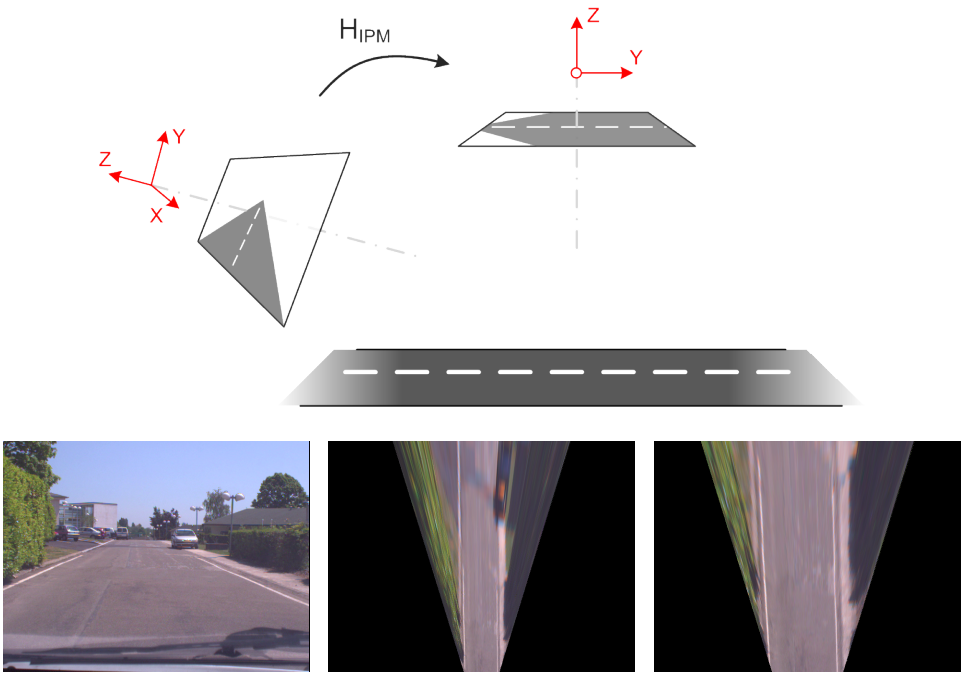


FIG. 5.9 – Transformation en coordonnées IPM sur 50 et 20 mètres. L'effet de perspective est annulé en tout point de l'espace navigable.

et l'homographie induite par le plan de l'espace navigable, annulant la perspective en tout point du sol, s'écrit finalement :

$$\mathtt{H}_{ipm} = (\mathtt{KA})^{-1}$$

5.4 Conclusion

L'hypothèse de planéité du sol permet de considérer un modèle exact et suffisamment contraint du déplacement apparent de l'espace navigable. L'homographie plane (de $\mathbb{R}^{3\times3}$) permet ainsi de décrire la relation liant la projection des points de l'espace libre entre deux acquisitions successives, indépendamment de leur profondeur dans la scène. Elle peut être estimée de manière robuste, à l'aide d'un ensemble de points correctement distribués dans l'image précédente, et de leurs correspondants dans l'image courante. Le résultat du *Tensor Voting* assure, à cet effet, la pertinence des couples de points choisis. Les paramètres de déplacement du véhicule $\{\mathtt{R}, \mathbf{t}, \mathbf{n}\}$, contenus dans la matrice homographique, sont obtenus par la

décomposition de cette dernière.

La mesure dense du flot optique offre la possibilité d'identifier l'espace navigable à l'aide d'un critère géométrique, correspondant à la distance séparant le mouvement apparent du modèle homographique estimé. L'information résultante s'avère supérieure à celle apportée par l'utilisation d'un critère photométrique, avec notamment l'identification de l'espace navigable et des obstacles adjacents, mobiles ou fixes. Un filtrage temporel, reposant en partie sur la mesure de confiance du flot optique, grâce au résultat du *Tensor Voting*, renforce la robustesse du processus.

Enfin, la définition du plan du sol dans le repère caméra, permet d'estimer un modèle texturé 3-D dense de l'espace navigable. Ce modèle est essentiel pour une part importante des applications en robotique mobile, notamment lorsqu'il s'agit d'établir la localisation d'un véhicule au sein d'un environnement précédemment exploré.

Chapitre 6

Identification des obstacles mobiles

Par dualité, tout élément de la scène n'appartenant pas à l'espace navigable peut être perçu comme un obstacle, ou à défaut, comme une zone inaccessible par un véhicule de transport urbain. Les résultats présentés au cours du chapitre 5, notamment Fig. 5.6 et 5.8, illustrent ainsi comment les régions de l'image, qui ne satisfont pas la transformation homographique induite par l'espace libre, indiquent la présence d'obstacles, qu'ils soient ou non statiques. Il demeure cependant nécessaire de différencier les deux cas de figure, de manière à permettre au module de commande d'un véhicule intelligent, d'estimer une trajectoire sûre.

Dans ce but, le chapitre 6 présente une solution, fondée sur la géométrie projective, pour distinguer les obstacles mobiles de l'environnement. Une première partie propose de s'affranchir du mouvement apparent induit par l'*ego-motion* du véhicule, grâce à l'étude du déplacement parallaxe issu de la décomposition du flot optique. Une seconde partie détaille ensuite comment estimer la position 3-D des points d'un obstacle mobile, circulant sur l'espace libre, d'après son image dans le plan focal. Enfin, une dernière section revient sur le processus de segmentation du champ de déplacement et, à l'aide d'une structure multi-résolutions, améliore l'identification des éléments mobiles de l'image.

6.1 Contraintes de rigidité

La segmentation des éléments mobiles de l'environnement est rendue difficile par le mouvement propre du véhicule. L'observation de la scène, dans le référen-

tiel caméra, ajoute au déplacement des points de $\mathbb{R}^3$, une composante égale à l'inverse de la transformation du capteur. Aussi est-il nécessaire de caractériser, dans l'image, le champ de déplacement induit par la seule progression du véhicule, afin d'être en mesure de discriminer les obstacles animés, parmi les éléments fixes de la scène. A cet effet, on propose de reprendre les travaux d'Irani et Anadan, sur l'étude de la géométrie parallaxe pour l'analyse d'une scène 3-D [95], et d'y adjoindre une méthode robuste pour le choix des primitives géométriques servant à calculer certaines contraintes de rigidité détaillées dans la suite du chapitre.

6.1.1 Décomposition "plan + parallaxe"

Soient $\mathbf{P} = (X,Y,Z)^T$ et $\mathbf{P}' = (X',Y',Z')^T$ les coordonnées cartésiennes d'un point P de l'espace 3-D, exprimées respectivement dans les repères liés aux caméras de centre optique C_1 et C_2. La transformation rigide associée au changement de référentiel entre ces caméras peut être écrite de la manière suivante :

$$\mathbf{P}' = \mathtt{R}^T\mathbf{P} + \mathbf{t}'_{12}\,, \tag{6.1}$$

avec $\mathbf{t}'_{12}$, le vecteur de translation $\overrightarrow{C_2C_1}$ dans le repère lié à la seconde caméra et $\mathtt{R}$, la matrice de rotation entre les référentiels des deux caméras.

Soient Π un plan quelconque de la scène observée, $\mathbf{n}$ sa normale (unitaire) dans le système de coordonnées associé à la première caméra et $\mathbf{n}'$ dans celui lié à la seconde. Pour tout point P appartenant à Π :

$$\mathbf{n}'^T\mathbf{P}' = d'_\pi \tag{6.2}$$

avec d'_π la distance du centre optique C_2 au plan Π. L'équation (6.2) peut être généralisée à tout point de la scène situé à une hauteur h du plan considéré, de la façon suivante :

$$\mathbf{n}'^T\mathbf{P}' = d'_\pi + h$$

ou encore :

$$\frac{\mathbf{n}'^T\mathbf{P}' - h}{d'_\pi} = 1 \tag{6.3}$$

En notant $\mathbf{t}'_{12}$ le vecteur $\overrightarrow{C_1C_2}$ exprimé dans le référentiel de la première caméra, $\mathbf{t}'_{12} = -\mathtt{R}^T\mathbf{t}_{21}$, l'équation (6.1) peut se réécrire :

$$\mathbf{P} = \mathtt{R}\mathbf{P}' - \mathtt{R}\mathbf{t}'_{12} = \mathtt{R}\mathbf{P}' + \mathbf{t}_{21}$$

et d'après l'équation (6.3), P devient :

$$\mathbf{P} = \mathtt{R}\mathbf{P}' + \mathbf{t}_{21}\frac{\mathbf{n}'^T\mathbf{P}' - h}{d'_\pi} = \left(\mathtt{R} + \frac{\mathbf{t}_{21}}{d'_\pi}\mathbf{n}'^T\right)\mathbf{P}' - \frac{h}{d'_\pi}\mathbf{t}_{21} \tag{6.4}$$

Soient $\mathbf{p} = (x,y,1)^T = \frac{1}{Z}\mathtt{K}\mathbf{P}$ et $\mathbf{p}' = (x,y,1)^T = \frac{1}{Z'}\mathtt{K}'\mathbf{P}'$, les coordonnées des points induits par la projection de P dans le plan rétinien de chacune des caméras. $\mathtt{K}$ et $\mathtt{K}'$ désignent les matrices des paramètres intrinsèques correspondant à ces caméras. On peut également définir la projection focale du vecteur de translation $\mathbf{t}_{21} = (T_x, T_y, T_z)^T$, telle que $\mathbf{t} = (t_x, t_z, T_z) = \mathtt{K}\mathbf{t}_{21}$. De cette façon, la multiplication des deux termes de l'équation (6.4) par $\frac{1}{Z}\mathtt{K}$ permet d'écrire :

$$\frac{Z}{Z'}\mathbf{p} = \mathtt{K}\left(\mathtt{R} + \frac{\mathbf{t}_{21}}{d'_\pi}\mathbf{n}'^T\right)\mathtt{K}'^{-1}\mathbf{p}' - \frac{h}{d'_\pi Z'}\mathbf{t} \tag{6.5}$$

d'où :

$$\mathbf{p} \propto \mathtt{A}'\mathbf{p}' - \frac{h}{d'_\pi Z'}\mathbf{t}$$

avec $\mathtt{A}'$ la matrice homographique induite par le plan Π, entre les prises de vue des caméras 1 et 2. L'égalité stricte est obtenue lorsque la troisième composante des coordonnées homogènes du terme de gauche est égale à 1, d'où :

$$\begin{aligned}
\mathbf{p} &= \frac{\mathtt{A}'\mathbf{p}' - \frac{h}{d'_\pi Z'}\mathbf{t}}{\mathbf{a}'_3\mathbf{p}' - \frac{h}{d'_\pi Z'}T_z} \\
&= \frac{\mathtt{A}'\mathbf{p}'}{\mathbf{a}'_3\mathbf{p}'} - \frac{\mathtt{A}'\mathbf{p}'}{\mathbf{a}'_3\mathbf{p}'} + \frac{\mathtt{A}'\mathbf{p}' - \frac{h}{d'_\pi Z'}\mathbf{t}}{\mathbf{a}'_3\mathbf{p}' - \frac{h}{d'_\pi Z'}T_z} \\
&= \frac{\mathtt{A}'\mathbf{p}'}{\mathbf{a}'_3\mathbf{p}'} + \frac{\frac{h}{d'_\pi Z'}T_z}{\left(\mathbf{a}'_3\mathbf{p}' - \frac{h}{d'_\pi Z'}T_z\right)}\frac{\mathtt{A}'\mathbf{p}'}{\mathbf{a}'_3\mathbf{p}'} - \frac{\frac{h}{d'_\pi Z'}\mathbf{t}}{\mathbf{a}'_3\mathbf{p}' - \frac{h}{d'_\pi Z'}T_z}
\end{aligned} \tag{6.6}$$

avec $\mathbf{a}'_3$ le troisième et dernier vecteur ligne de la matrice $\mathtt{A}'$. La troisième composante du terme de gauche, dans l'équation vectorielle (6.5), peut donc s'écrire :

$$\frac{Z}{Z'} = \mathbf{a}'_3\mathbf{p}' - \frac{h}{d'_\pi Z'}T_z$$

Par substitution dans (6.6), on obtient alors :

$$\mathbf{p} = \frac{\mathtt{A}'\mathbf{p}'}{\mathbf{a}'_3\mathbf{p}'} + \frac{h}{Z}\frac{T_z}{d'_\pi}\frac{\mathtt{A}'\mathbf{p}'}{\mathbf{a}'_3\mathbf{p}'} - \frac{h}{Zd'_\pi}\mathbf{t}$$

Lorsque $T_z \neq 0$, l'épipôle situé dans le plan focal de la première caméra, à savoir l'image du centre optique C_2 vu par cette dernière, est défini de la manière suivante :

$$\mathbf{e} = \frac{1}{T_z}\mathtt{K}\mathbf{t}_{21} = \frac{1}{T_z}\mathbf{t}$$

D'où :

$$\mathbf{p} = \frac{\mathtt{A}'\mathbf{p}'}{\mathbf{a}'_3\mathbf{p}'} + \frac{h}{Z}\frac{T_z}{d'_\pi}\left(\frac{\mathtt{A}'\mathbf{p}'}{\mathbf{a}'_3\mathbf{p}'} - \mathbf{e}\right) \tag{6.7}$$

Le vecteur $\dfrac{\mathtt{A}'\mathbf{p}'}{\mathbf{a}'_3\mathbf{p}'}$ qui définit les coordonnées résultant de la transformation homographique de $\mathbf{p}'$, décrite par la matrice $\mathtt{A}'$, est noté $\mathbf{p}_w$. Par substitution dans l'équation (6.7), on obtient donc :

$$\mathbf{p} = \mathbf{p}_w + \gamma\frac{T_z}{d'_\pi}(\mathbf{p}_w - \mathbf{e}) \tag{6.8}$$

en posant $\gamma = h/Z$, la *structure projective* 3-D du point P par rapport au plan Π. Lorsque P appartient à Π, $\mathbf{p} = \mathbf{p}_w$; dans le cas contraire, le déplacement résiduel est proportionnel à la structure projective et correspond à la parallaxe de mouvement, brièvement abordée dans la section 4.2.2. Le cas particulier $T_z = 0$ correspond à un véhicule statique. Les obstacles mobiles sont alors identifiés dans l'image par l'étude des régions de mouvement non nul.

En considérant, non plus les acquisitions de deux caméras distinctes, mais celles, consécutives, d'une seule caméra, l'équation (6.8) peut être reformulée de manière à exprimer le mouvement apparent $\mathbf{u}$, comme la somme d'une transformation homographique $\mathbf{u}_\pi$ et d'un *déplacement résiduel parallaxe* noté μ :

$$\underbrace{\mathbf{p}' - \mathbf{p}}_{\mathbf{u}} = \underbrace{\mathbf{p}' - \mathbf{p}_w}_{\mathbf{u}_\pi} \underbrace{-\gamma\frac{T_z}{d'_\pi}(\mathbf{p}_w - \mathbf{e})}_{\mu} \tag{6.9}$$

Cette décomposition permet, à l'aide du modèle homographique précédemment estimé, de calculer le mouvement résiduel parallaxe.

6.1.2 Contrainte relative de structure

Obstacles mobiles

Soient $\mathbf{p}_i = (x_i, y_i, 1)^T$, $i \in \{1;2\}$ les coordonnées de deux points appartenant à l'environnement statique observé. Le mouvement parallaxe correspondant est donné par :

$$\mu_i = \gamma_i \frac{T_z}{d'_\pi} (\mathbf{e} - \mathbf{p}_{wi})$$

d'où :

$$\mu_1 \gamma_2 - \mu_2 \gamma_1 = \gamma_1 \gamma_2 \frac{T_z}{d'_\pi} (\mathbf{p}_{w2} - \mathbf{p}_{w1})$$

Le terme $\gamma_1 \gamma_2 \dfrac{T_z}{d'_\pi}$ désigne un scalaire, les vecteurs $(\mu_1 \gamma_2 - \mu_2 \gamma_1)$ et $\triangle \mathbf{p}_w = (\mathbf{p}_{w2} - \mathbf{p}_{w1})$ sont donc colinéaires. Aussi, en notant $(\triangle \mathbf{p}_w)_\perp$ le vecteur orthogonal à $\triangle \mathbf{p}_w$:

$$(\mu_1 \gamma_2 - \mu_2 \gamma_1)^T (\triangle \mathbf{p}_w)_\perp = 0$$

et :

$$\frac{\gamma_2}{\gamma_1} = \frac{\mu_2^T (\triangle \mathbf{p}_w)_\perp}{\mu_1^T (\triangle \mathbf{p}_w)_\perp} \tag{6.10}$$

$\dfrac{\gamma_2}{\gamma_1}$ définit alors la *contrainte de structure relative* des points p_1 et p_2, invariante dans le temps pour toute paire de points 3-D appartenant à la même structure rigide. Son écriture sur trois images successives i, j et k, est donnée par :

$$\frac{\gamma_2}{\gamma_1} = \frac{\mu_2^{i,jT} (\triangle \mathbf{p}_w)_\perp^{i,j}}{\mu_1^{i,jT} (\triangle \mathbf{p}_w)_\perp^{i,j}} = \frac{\mu_2^{j,kT} (\triangle \mathbf{p}_w)_\perp^{j,k}}{\mu_1^{j,kT} (\triangle \mathbf{p}_w)_\perp^{j,k}} \tag{6.11}$$

en notant $\mu^{i,j}$ et $\triangle \mathbf{p_w}^{i,j}$ les vecteurs de déplacement μ et $\triangle \mathbf{p}_w$ entre les images i et j. L'équation (6.10) appliquée aux points P_1 et P_2, entre chaque paire d'images $\{i, j\}$ et $\{j, k\}$, permet d'écrire :

$$\left(\mu_1^{j,kT} (\triangle \mathbf{p}_w)_\perp^{j,k}\right) \left(\mu_2^{i,jT} (\triangle \mathbf{p}_w)_\perp^{i,j}\right) - \left(\mu_1^{i,jT} (\triangle \mathbf{p}_w)_\perp^{i,j}\right) \left(\mu_2^{j,kT} (\triangle \mathbf{p}_w)_\perp^{j,k}\right) = 0 \tag{6.12}$$

Il suffit ainsi de choisir le point de référence P_1 de sorte que son déplacement parallaxe soit non nul, pour évaluer la contrainte (6.12) en tout point présent à la fois dans les images i, j et k. De cette façon, on peut identifier l'ensemble des points 3-D appartenant à la structure rigide de P_1. Lorsque ce dernier est statique, cela revient à différencier les parties fixes et mobiles de la scène observée.

L'espace navigable, tel qu'il est déterminé dans la section 5.2.2, définit l'unique plan de la scène dont on connaît le modèle et garantit le caractère statique des points 3-D qui le constituent. Il est toutefois impossible de choisir un point de référence appartenant au plan de l'espace libre pour calculer la structure relative induite par ce dernier. Dans ce cas, en effet, le déplacement parallaxe μ_1 serait nul, aux imprécisions près du modèle homographique. On propose donc d'utiliser les paramètres de déplacement de la caméra $\left\{\mathtt{R}, \frac{\mathbf{t}}{d}, \mathbf{n}\right\}$, obtenus après décomposition du mouvement image induit par l'espace libre, afin de calculer l'homographie d'un plan virtuel auquel n'appartient aucun des points du sol. On considère donc le plan à l'infini Π_∞, orthogonal à l'axe focal du capteur photographique. D'après l'équation (5.3), le modèle homographique induit par Π_∞ dépend uniquement de la composante rotationnelle du mouvement caméra :

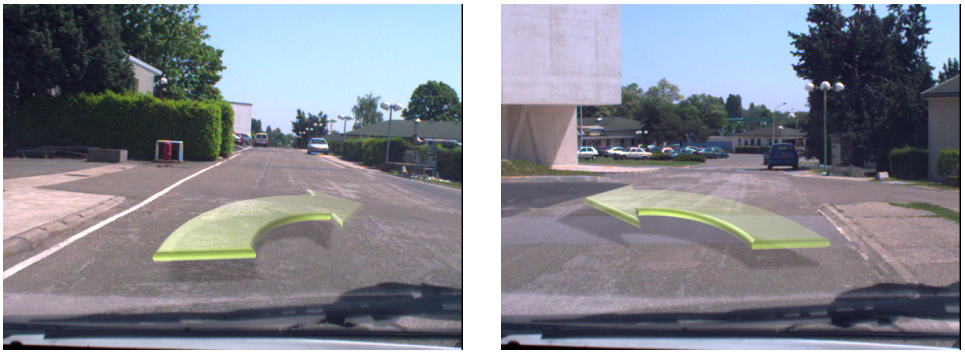
$$\mathtt{H}_\infty = \mathtt{K}\mathtt{R}\mathtt{K}^{-1}$$

$\mathtt{H}_\infty$ définit le champ de déplacement visuel des éléments constituant l'arrière-plan de la scène. Le mouvement résiduel parallaxe, calculé d'après la décomposition (6.9), correspond alors à la composante translationnelle du déplacement apparent (Fig. 6.1).

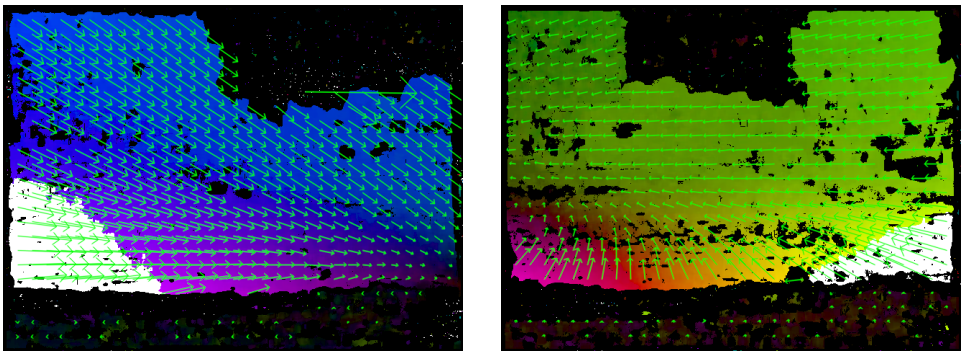
On sélectionne donc un point de référence p_1, appartenant au plan de l'espace navigable, suffisamment proche du centre optique pour ne pas être confondu avec le plan à l'infini. De cette façon, le point de référence est nécessairement statique. En outre, plutôt que de calculer la *contrainte de structure relative* associée à ce point, à partir d'un mouvement image estimé, on évalue cette contrainte à l'aide du modèle de déplacement de l'espace libre. Le mouvement parallaxe μ_1 du point de référence est ainsi calculé grâce à l'homographie induite par la translation de la caméra entre deux acquisitions :

$$\mu_1 = \mathtt{H}_\mathbf{t}\mathbf{p}_1 = \mathtt{K}\left(\frac{\mathbf{t}}{d}\mathbf{n}^T\right)\mathtt{K}^{-1}\mathbf{p}_1$$

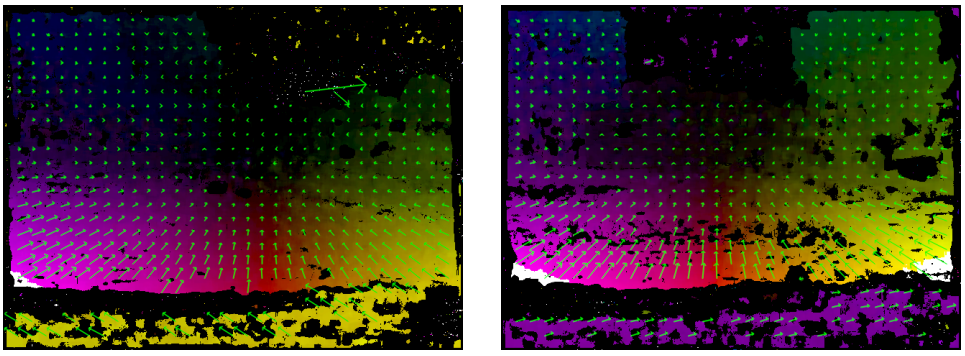
Enfin, puisque ce modèle est défini en tout point de $\mathbb{P}^2$, quelles que soient ses coordonnées, le choix de p_1 ne dépend plus, ni de la projection de l'espace navigable dans le plan focal, ni des limites spatiales de l'image. En sélectionnant p_1 de sorte que la norme du mouvement parallaxe correspondant soit non nulle (généralement l'un des coins inférieurs de l'image) la *contrainte de structure relative* peut être calculée pour tous les autres points du plan focal. Toutefois, dans la direction



Images d'entrée.



Flot optique combiné par Tensor Voting.



Mouvement résiduel parallaxe. Le champ de déplacement est radial : l'ensemble des vecteurs associés aux points fixes de la scène convergent vers l'épipôle.

FIG. 6.1 – Estimation du mouvement résiduel parallaxe pour deux séquences témoins (colonnes droite et gauche). Les champs de déplacement calculés sont représentés à l'aide du code couleur détaillé au chapitre 3 ainsi que de façon éparse, à l'aide de flèches dessinées en sur-impression..

du vecteur de déplacement parallaxe μ_1, $\triangle \mathbf{p}_w$ est nul. Il est donc nécessaire d'estimer la contrainte de rigidité à l'aide d'un second point de référence (par exemple, le coin inférieur restant de l'image), sur et à proximité de l'axe du déplacement parallaxe de p_1.

La *contrainte de structure relative* est suffisante lorsque l'axe du déplacement résiduel des obstacles mobiles ne converge pas vers le point d'expansion de l'image, ou épipôle. Elle permet, par exemple, d'identifier tout objet se déplaçant orthogonalement à l'axe optique. En revanche, elle n'est plus suffisante dès lors qu'un obstacle se déplace le long de cet axe. Il est alors nécessaire d'étudier, non plus l'orientation du déplacement parallaxe, mais sa norme.

Carte de profondeurs et consistance 3-D

Pour identifier les éléments mobiles de la scène, notamment ceux circulant le long de l'axe optique, on propose, dans la suite du chapitre, d'estimer la profondeur de chaque élément comme s'il était statique, à l'aide de l'équation 6.11, puis de comparer cette mesure avec la profondeur correspondant au point de contact de cet élément avec le sol, grâce au modèle homographique précédemment estimé. Ces deux estimations, équivalentes dans le cas d'un objet statique, sont inconsistantes dans le cas d'un obstacle mobile.

La *contrainte de structure relative* $\dfrac{\gamma_2}{\gamma_1}$, définie par l'équation (6.11), correspond à la profondeur relative du point P_1 par rapport à celle du point P_2. La valeur de cette contrainte, en tout point de l'image, permet de dessiner une carte des profondeurs relatives, analogue à la carte des disparités généralement estimée en stéréovision. La figure 6.2 illustre la carte des profondeurs calculée pour trois paires d'images distinctes, tirées de la séquence *Route*. Deux points de référence, P_1et P_2, correspondant aux coins inférieurs droit et gauche de l'image, ont été sélectionnés pour éviter les singularités dans la direction de leur déplacement parallaxe respectif. En tout point p_i de l'image, la profondeur est néanmoins estimée par rapport à P_1, quel que soit le point de référence utilisé, avec $\dfrac{\gamma_i}{\gamma_1} = \dfrac{\gamma_2}{\gamma_1}\dfrac{\gamma_i}{\gamma_2}$. Les cartes de profondeur ainsi obtenues décrivent distinctement le relief de la scène observée. Toutefois, dans le cas des obstacles mobiles, et malgré le faible déplacement de ces derniers, on note une différence entre, d'une part, la profondeur de l'espace navigable au point de contact des obstacles avec le sol et, d'autre part, la profondeur moyenne estimée pour les obstacles eux-même. Cette inconsistance structurelle autorise l'identification des obstacles mobiles, quelle que soit la nature de leur déplacement sur le plan

de l'espace navigable.

La profondeur réelle, des éléments fixes de la scène, peut être obtenue à partir de leur structure relative par rapport au point de référence, dès lors qu'est connue la profondeur de ce dernier. En outre, l'ensemble des équations nécessaires au calcul de la position des points de référence, à partir du modèle homographique induite par le plan du sol, est détaillé dans la section suivante.

Le modèle minimal étendu, défini au chapitre 2, ne considère pas le cas des obstacles statiques suspendus, tels que les barrières de péages. Rares en milieu urbain, ces obstacles sont généralement ignorés lors des études sur les véhicules automatisées. Dans le cadre des travaux présentés, ils ne sont donc pas traités directement. Toutefois, l'image d'un objet suspendu est interprétée comme un obstacle mobile, posé sur la route au niveau du point d'intersection avec le rayon optique traversant la base de l'objet. La position virtuelle d'un tel obstacle, converge dans le temps, avec le rapprochement du véhicule, vers la position de l'objet réel. Mal classifiés, ces éléments sont donc néanmoins perçus, de sorte qu'il est possible de les éviter.

6.2 Projection inverse des points de $\mathbb{P}^2$

D'après la section 2.2.2, la projection inverse des points de l'image conduit au système sous-dimensionné (2.8) :

$$\begin{cases} X = Z(u-u_0)/fk_u \\ Y = Z(v-v_0)/fk_v \end{cases} \tag{6.13}$$

avec $(u,v)^T$ la position dans l'image d'un point 3-D de coordonnées $(X,Y,Z)^T$. Le centre de l'image est donné par $(u_0,v_0)^T$ tandis que fk_u et fk_v correspondent aux paramètres intrinsèques du capteur photographique. Pour contraindre ce système, une solution consiste à étudier l'homographie correspondant à la projection inverse des points de l'image sur un plan de $\mathbb{R}^3$. Soit Π ce plan et $\mathbf{n} = (n_x,n_y,n_z)^T$ une normale de Π. Pour tout point M du plan et de coordonnées $(X,Y,Z)^T$, la distance de Π jusqu'au centre du repère caméra s'écrit alors :

$$d_\pi = n_x X + n_y Y + n_z Z \tag{6.14}$$

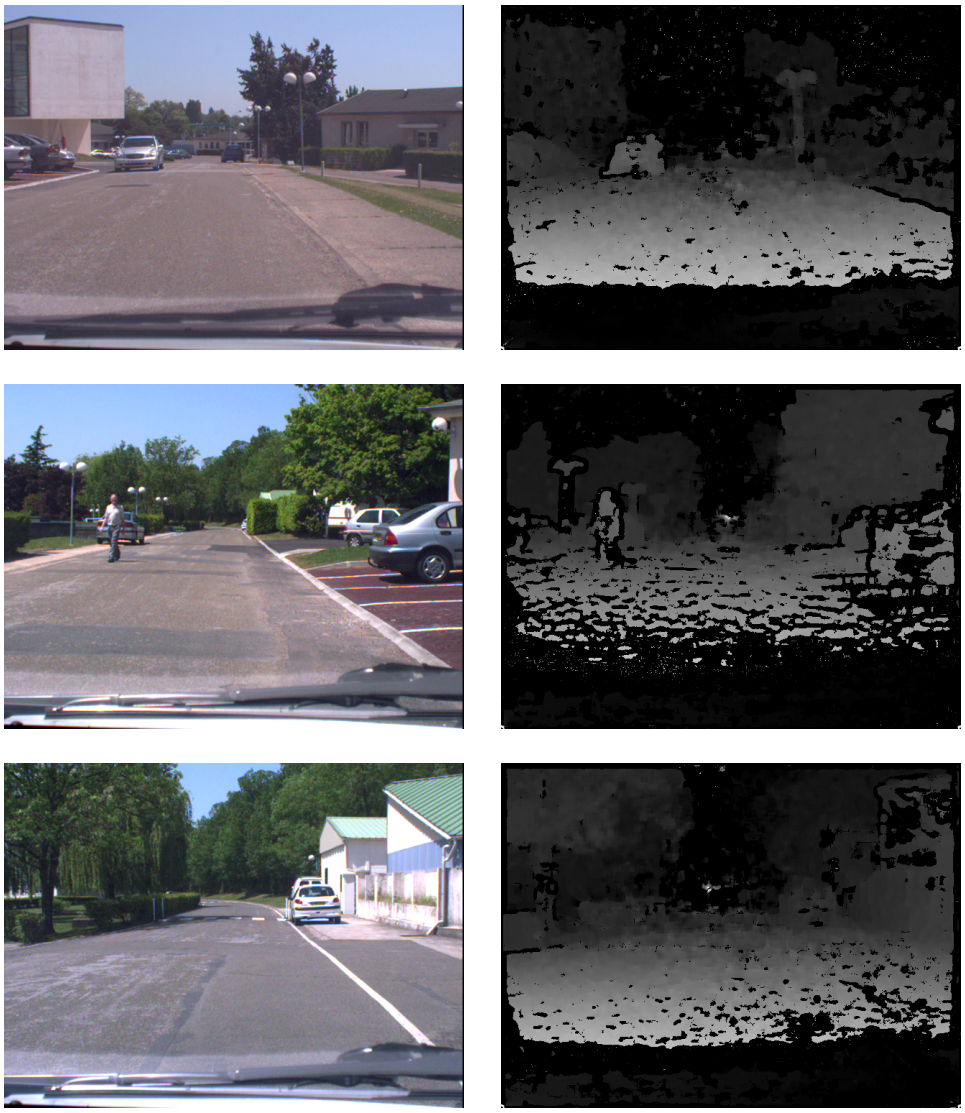


FIG. 6.2 – Carte des profondeurs relatives. Du plus clair au plus sombre, dans le sens des profondeurs croissantes (en noir, les profondeurs non estimées).

En substituant dans l'équation (6.14), l'expression des composantes X et Y, donnée par le système (6.13), on obtient :

$$d = Z\left(\frac{n_x(u-u_0)}{fk_u} + \frac{n_y(v-v_0)}{fk_v} + n_z\right)$$

En notant $\psi = \frac{n_x(u-u_0)}{fk_u} + \frac{n_y(v-v_0)}{fk_v} + n_z$, les coordonnées du point M vérifient alors :

$$\begin{cases} Z &= d/\psi \\ X &= d(u-u_0)/(fk_u\psi) \\ Y &= d(v-v_0)/(fk_v\psi) \end{cases} \tag{6.15}$$

Il est ainsi possible d'estimer la position de tout point de l'espace, d'après sa projection dans l'image et la définition d'un plan de $\mathbb{R}^3$ auquel il appartient. On emploie donc le système (6.15), appliqué au plan de l'espace navigable précédemment défini, pour évaluer, par exemple, la profondeur d'un obstacle mobile au niveau du point de contact avec l'espace navigable.

6.3 Segmentation multi-échelle de l'image

Le partitionnement du flot optique, telle qu'il est présenté au chapitre 3, nécessite de fixer plusieurs seuils afin de filtrer les minimas locaux qui conduiraient, sans cette étape, à la sur-segmentation de l'image. Chaque seuil est associé à un critère géométrique en relation avec la topographie du signal 2-D correspondant au résultat du *Tensor Voting*, l'image des discontinuités du flot optique. Le filtrage est exécuté à l'aide d'une structure hiérarchique formant une représentation multi-résolutions de l'image segmentée : l'arbre des composantes ou *Min-Tree*.

La difficulté du seuillage provient des disparités d'amplitude parmi les discontinuités du mouvement apparent sur lesquelles s'appuie le processus de segmentation. Tout comme la superficie des obstacles dans l'image, les discontinuités associées au mouvement relatif de ces derniers diminuent avec leur éloignement dans la scène. Il est donc nécessaire d'employer des seuils adaptatifs, fonctions de la profondeur des cellules considérées dans l'image. Par ailleurs, les discontinuités du flot optique sont parfois insuffisantes pour induire une véritable rupture, au sein de l'espace 4-D (x, y, v_x, v_y), au niveau du contour entre les des obstacles mobiles et l'espace navigable. La figure 6.3 illustre cette situation. Le déplacement apparent du véhicule, au point de contact avec le sol, y est égal au mouvement image

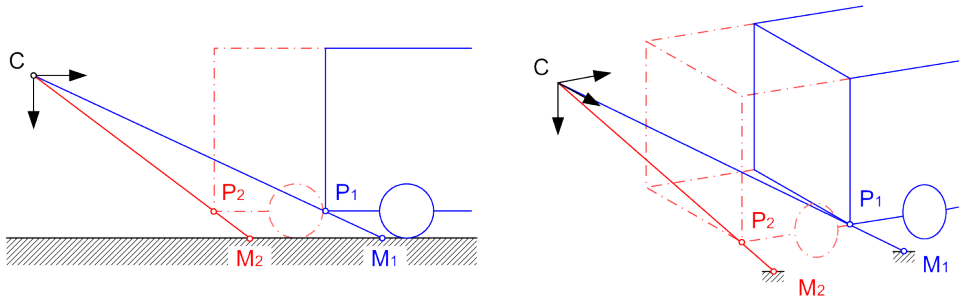


FIG. 6.3 – Lorsque le mouvement image d'un véhicule est localement identique au déplacement apparent de la partie du sol qu'il masque, il n'existe plus de discontinuité spatiale entre l'obstacle et le sol, dans l'espace 4-D (x, y, v_x, v_y).

de la partie masquée de l'espace navigable : la projection focale du vecteur $\overrightarrow{P_1P_2}$, exprimant la combinaison de l'*ego-motion* et du déplacement propre du point P, est identique à celle du vecteur $\overrightarrow{M_1M_2}$, qui traduit le mouvement de la caméra dans son propre repère. Les hypersurfaces correspondant au champ de déplacement visuel de l'espace libre, d'une part, et au mouvement apparent de l'obstacle étudié, d'autre part, sont alors connexes dans l'espace (x, y, v_x, v_y). La carte des discontinuités qui résulte du *Tensor Voting* 4-D n'assure plus alors la segmentation de l'obstacle à l'issue du *watershed*.

On remédie à chacun des problèmes énoncés en modifiant la construction de l'arbre des composantes à l'aide de différentes contraintes. Un premier critère, sur la profondeur des points de l'image, permet de ne pas fusionner les noeuds de l'arbre correspondant aux régions connexes dans le plan focal mais dont la distance à la caméra diffère. De même, un critère sur la distance, entre le flot optique et le modèle homographique induit par l'espace navigable, permet d'assurer une rupture, à la frontière entre le sol et les obstacles mobiles.

La structure ainsi obtenue se compose d'un ensemble d'arbres disjoints, correspondant chacun à une partie de l'image, spatialement et dynamiquement homogène. En reliant leur racines à un noeud commun, la représentation finale s'apparente conceptuellement à un arbre scénique tel que ceux pouvant être utilisés dans la modélisation de scènes 3-D en infographie. Comme pour les opérateurs connexes décrits au cours du chapitre 3, la segmentation du flot optique est alors réalisée en parcourant l'arbre scénique.

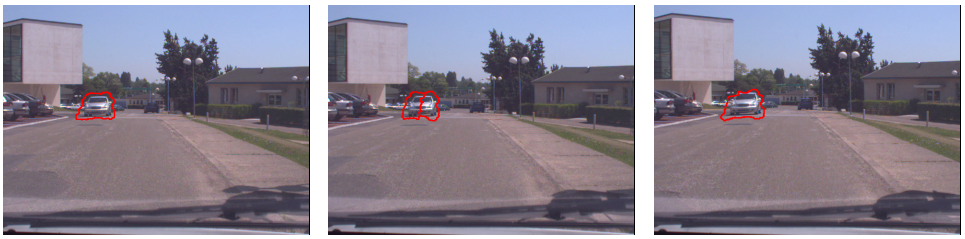


FIG. 6.4 – Identification d'un obstacle mobile sur 3 images consécutives à partir de la segmentation du flot optique par LPE ainsi que du modèle homographique induit par le plan du sol.

6.4 Conclusion

La seule définition de l'espace navigable, par la décomposition de son modèle homographique, permet de classifier l'ensemble des éléments de la scène, selon qu'ils soient statiques ou mobiles. Tandis que les principales approches, développées à cet effet, reposent sur la convergence du mouvement résiduel parallaxe vers l'épipôle, à l'aide des contraintes de rigidité classiques, on propose d'étudier la norme de ce mouvement à travers un critère de consistance structurelle. On suppose pour cela que le déplacement des obstacles mobiles est connexe au plan du sol.

L'intégration de la mesure de profondeur ainsi que de l'erreur de reprojection du modèle homographique induit par l'espace navigable, améliore la robustesse du processus de segmentation présenté au chapitre 3, et notamment l'étape de filtrage. En outre, la construction de l'arbre des composantes donne ainsi une représentation des éléments de l'image se rapprochant des graphes scéniques traditionnellement utilisés pour manipuler un modèle 3-D.

Quatrième partie

Intégration

Chapitre 7

Solutions techniques et conclusions

Parmi toutes les étapes de la construction du modèle minimal étendu de la scène, certains processus ne peuvent être exécutés en temps réel sur les architectures mono-processeur actuelles. L'estimation du flot optique ainsi que le *Tensor Voting* ont une complexité algorithmique élevée du fait qu'ils nécessitent de parcourir le voisinage de chaque pixel de l'image. Toutefois, et pour les mêmes raisons, ces processus sont hautement parallélisables.

Après avoir présenté, dans une première section, les principales solutions *multi-cores* adaptées au calcul intensif, la suite du chapitre détaille l'architecture CUDA, de la firme NVIDIA, et conclut par le résultat de l'implémentation des algorithmes d'estimation du flot optique et de *Tensor Voting* sur carte graphique.

7.1 Différentes architectures

7.1.1 Les réseaux logiques programmables

Les réseaux logiques programmables, ou FPGA[1], sont des circuits composés de nombreuses cellules logiques élémentaires, connectées de manière réversible par programmation. Il est souvent nécessaire d'utiliser un langage de description matériel, tel que HDL, pour concevoir un circuit électronique. Le compilateur se charge alors de convertir le programme en un schéma logique inter-connectant les différentes cellules utiles du FPGA. Toutefois, la conception d'une chaîne de trai-

[1] ***F**ield-**P**rogrammable **G**ate **A**rray.*

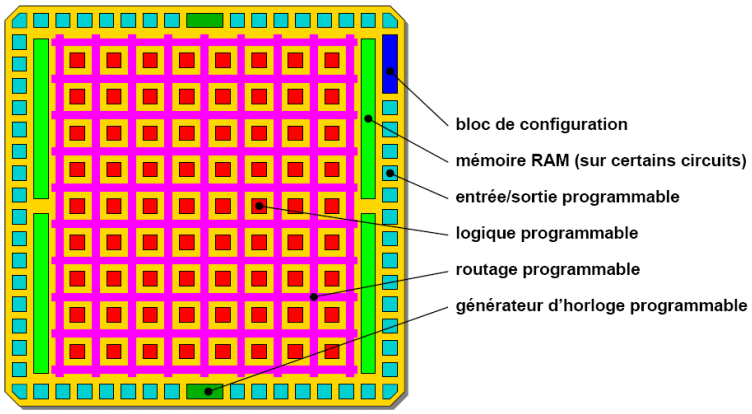


FIG. 7.1 – Schéma d'un réseau logique programmable (FPGA).

tement, du signal d'entrée, se révèle d'autant plus fastidieuse que la recherche d'erreurs et les temps de compilation sont particulièrement longs. En outre, et malgré le caractère générique des circuits de réseaux logiques programmables, le prix de ces derniers est nettement supérieur à celui d'un GPU. Pour ces raisons, les FPGAs sont principalement réservés à la mise en oeuvre de systèmes embarqués, contraints en terme d'espace et de consommation d'énergie.

7.1.2 Le GPGPU

Le terme GPGPU [2] désigne l'utilisation des unités de calcul graphique comme co-processeurs spécialisés dans le traitement de tâches hautement parallélisables. Ces unités de calcul sont généralement localisées sur la carte d'extension vidéo des ordinateurs individuels, mais peuvent également appartenir à un carte "graphique" dédiée au calcul intensif, à ce titre dénuée de sortie vidéo.

GPGPU : le calcul par l'affichage

L'histoire du GPGPU débute réellement dans les années 2000 avec l'apparition des chaînes de traitement graphique, ou *pipelines,* programmables. Les langages de programmation haut-niveaux, tels que Cg (2002), HLSL et GLSL (2003), ajoutent une couche d'abstraction au-dessus des interfaces de programmation (API) graphique existantes (OpenGL, DirectX, etc.). L'exploitation des ressources GPU nécessite néanmoins d'intégrer les algorithmes dans le processus d'affichage. Les

[2] ***G****eneral-purpose* ***P****rocessing on* ***G****raphics* ***P****rocessing* ***U****nits.*

tableaux deviennent des textures, les calculs des opérations de rendu. La spécialisation du *pipeline* graphique et des unités de calcul, réparties entre les processeurs de sommets et de fragments, complique notamment l'accès à la mémoire ainsi que la conception des programmes et leur maintenance. Le GPGPU profite alors essentiellement au domaine de la simulation physique.

NVIDIA CUDA et ATI Stream

Depuis novembre 2006 et le *chipset* graphique Nvidia G80, les architectures unifiées, composées de processeurs de flux non spécialisés de type SIMD[3], ont progressivement remplacé les architectures traditionnelles au sein des GPU. On définit le flux comme un jeu d'éléments indépendants, généralement structurés spatialement dans un grille, qui requièrent un traitement similaire. Le traitement appliqué à chaque élément du flux est appelé noyau, ou *kernel*.

Les deux principaux manufacturiers de cartes graphiques distribuent aujourd'hui leur propre solution GPGPU : Stream SDK pour ATI, CUDA pour NVIDIA. Ces technologies comprennent :

- un langage haut niveau (CUDA et Brook+) pour exploiter simplement les ressources GPU,
- une API bas niveau (PTX et CAL) pour permettre aux développeurs d'accéder explicitement à toutes les opérations GPU,
- un pilote avec lequel interagit l'API bas niveau,
- et un ensemble d'outils et de librairies facilitant le développement.

La puissance de calcul d'un processeur se mesure en nombre d'opérations flottantes par seconde (flops). Dans le cas des GPU, l'ensemble du circuit graphique et des processeurs de flux qui la composent estt considéré comme un unique coprocesseur. C'est sur cette base qu'il est possible de comparer les performances entre GPU et CPU. La figure 7.2 illustre ainsi la progression des *chipsets* graphiques NVIDIA face aux processeurs Intel, de 2003 à 2008. Dans ce laps de temps, la croissance des performances des GPU a été cinq fois supérieure à celle des CPU. A prix équivalent, la production CPU actuelle propose un maximum de 4 coeurs fonctionnant à 3.2 GHz, contre 240 processeurs de flux à 1.3 GHz pour une carte NVIDIA GT200. Avec une puissance de calcul de près d'un teraflops, les cartes graphiques de dernière génération, qui s'apparentent à des processeurs vectoriels, représentent donc un moyen économique de posséder un supercalculteur.

[3] ***S**ingle **I**nstruction **M**ultiple **D**ata.*

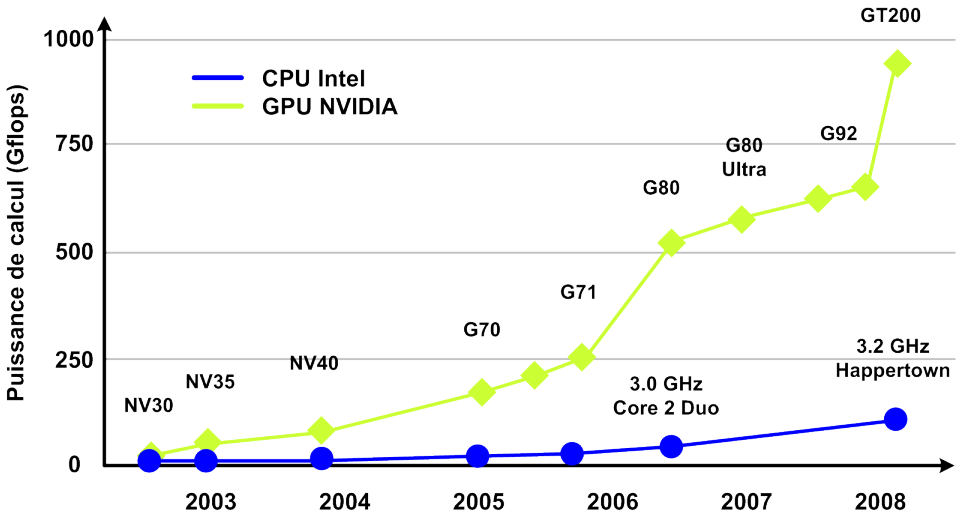


FIG. 7.2 – Evolution de la puissance de calcul (en Gflops) des CPU et GPU de 2003 à 2008.

Intel Larrabee

Concurrencé par les constructeurs de cartes graphiques sur le secteur des systèmes de calcul intensif, Intel développe actuellement une architecture multi-coeurs de type x86, baptisée *Larrabee* [96]. Cette dernière intègre 32 processeurs de type *Pentium P54C*, pour chacun desquels le rapport performance/superficie surpasse celui des CPU de dernière génération. L'avantage d'une telle plate-forme, sur les solutions concurrentes, réside essentiellement dans l'accessibilité du modèle de programmation. Larrabee utilise le jeu d'instructions x86 standard, auquel vient s'ajouter un certains nombre d'extensions qui lui sont spécifiques. La compilation est réalisée à l'aide du compilateur *Larrabee Native's C/C++*, intégrant un jeu d'instructions étendu. Aussi, la majeure partie des applications C/C++ courante peut être exécutée sur Larrabee, sans nécessitér d'autre modification que la recompilation du code source.

Les performances annoncées par Intel sont de l'ordre de 2 billions d'opérations par seconde (2 Tflops) pour une consommation d'environ 300W. Non commercialisée avant la fin 2009, le *chipset* Larrabee préfigure toutefois les performances dont il faut déjà tenir compte, pour penser les applications temps-réel de demain.

7.2 CUDA

L'accessibilité du langage CUDA et le support fourni par une communauté croissante et réactive de développeurs sont les principales raisons qui ont conduit à choisir la solution de la firme NVIDIA, plutôt qu'une technologie concurrente. La suite de cette section détaille donc certaines spécificités du développement CUDA, avant de présenter les résultats obtenus pour l'intégration des algorithmes de calcul du flot optique et de *Tensor Voting*.

7.2.1 Architecture unifiée NVIDIA

Architecture matérielle

Depuis CUDA, l'ensemble de la production NVIDIA adopte une architecture commune, unifiée, de façon à assurer la pérennité des développements. Chaque circuit graphique est ainsi composé d'un ensemble de multiprocesseurs, constitués de plusieurs processeurs de flux 32 bits. Au sein d'un même multiprocesseur, ces unités de calcul de type SIMD exécutent, en parallèle, un jeu d'instructions identique sur le flux de données. Des restrictions techniques, concernant le partage des ressources mémoire, sont à l'origine de la division de l'ensemble des processeurs de flux en plusieurs multiprocesseurs.

La figure 7.3 présente l'agencement des différents types de mémoire, globale d'une part, propre à chaque multiprocesseur d'autre part. Les échanges mémoire avec les unités de calcul sont modélisés par des flèches de couleur, indiquant non seulement le sens du transfert (lecture, écriture ou les deux), mais également le temps de latence induit : 1 cycle d'horloge pour les flèches vertes, entre 2 et 300 cycles pour les rouges. L'accès aux modules de mémoire embarquée des multiprocesseurs est rapide, tandis que la mémoire commune, plus abondante, est plus lente. Toute la difficulté d'intégration d'un algorithme consiste donc à gérer les ressources disponibles, en optimisant l'utilisation de la mémoire locale, très limitée mais performante, et à réduire les accès à la mémoire globale, aux seules lectures des données et écritures des résultats.

Modèle de programmation

Pour les raisons techniques précédemment abordées, mais également pour faciliter la gestion des ressources GPU, la programmation CUDA suit un modèle hiérarchique. Les *threads*, qui correspondent à la séquence d'instructions associée

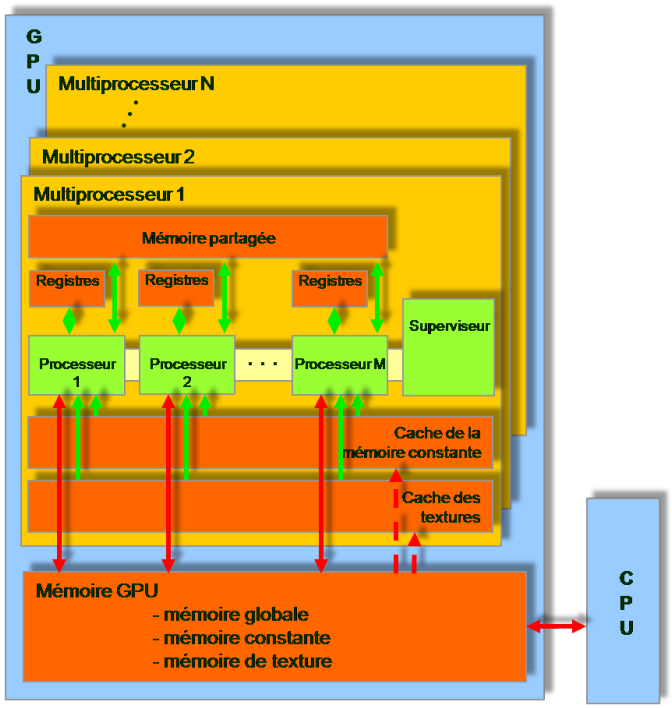


FIG. 7.3 – Organisation des différents modules de mémoire au sein des GPU NVIDIA.

à chaque élément du flux, sont regroupés en plusieurs *blocs*, eux-même agencés en *grille* (Fig. 7.4). Plusieurs blocs de *threads* sont assignés de façon définitive à chaque multiprocesseur, afin d'être exécutés concurremment, en temps partagé, pour cacher la latence des accès à la mémoire globale. Plus la taille d'un bloc est importante, plus les ressources locales attribuées à chaque *thread* sont faibles. A l'inverse, le nombre de *threads* doit être suffisant pour garantir un taux d'occupation maximal des processeurs de flux. Il ne suffit toutefois pas qu'un bloc compte plus de *threads* que d'unités de calcul pour cela. Les instructions conditionnelles sont la source de divergences, lorsqu'une partie seulement des *threads* doit exécuter certaines instructions. Les processeurs de flux assignés aux *threads* non concernés sont alors inactifs jusqu'à la reprise du code commun. Enfin, le découpage en grilles est une conséquence de la limitation du nombre de *threads* par bloc ainsi que du nombre de blocs gérés simultanément par le GPU.

Il existe certains problèmes, dont notamment le traitement d'images, pour lesquels la localité spatiale des données peut être exploitée. Aussi le modèle de programmation CUDA propose-t-il d'agencer les blocs et les grilles, en tableaux 1-D,

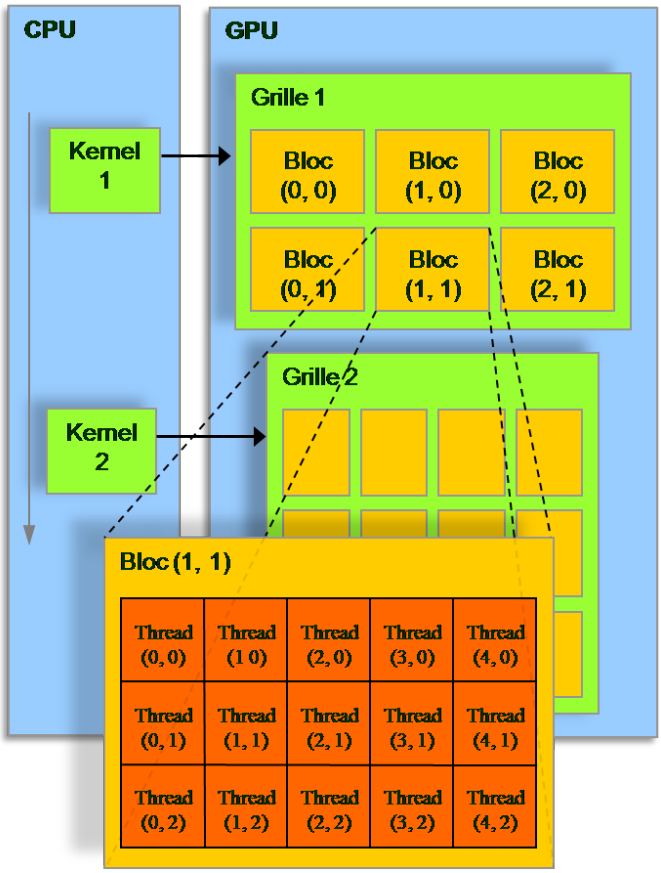


FIG. 7.4 – Modèle de programmation GPU

2-D ou 3-D. Il est ainsi d'autant plus simple, par exemple, de faire correspondre chaque *thread* à un pixel d'une image, quand le traitement de ce dernier est indépendant de celui de ces voisins.

Pour conclure, l'exécution des *kernels* est séquentielle mais pas nécessairement synchrone. En d'autres mots, un GPU ne peut prendre en charge plus d'un *kernel* à la fois, mais l'appel de ce dernier n'est pas bloquant (sous certaines conditions) pour le CPU appelant. Une architecture multi-GPU offre la possibilité de lancer plusieurs *kernels* en parallèle, à raison d'un par GPU et par *thread* CPU. Une présentation plus exhaustive de la technologie CUDA [97] est disponible sur le site de la firme NVIDIA `http://www.nvidia.com/cuda`.

7.2.2 Intégration

Estimation du flot optique

L'algorithme d'estimation du flot optique reprend la solution pyramidale de Lucas et Kanade présentée au chapitre 3. Afin de limiter les accès à la mémoire globale aux seules lecture des données et écriture des résultats, le programme est découpé en quatre *kernels* ayant pour tâches respectives, la constructions des pyramides, le calcul des dérivées, l'estimation du flot optique à chaque itération et l'interpolation du champ de déplacement entre chaque niveau des pyramides. Le diagramme de la figure 7.5 décrit l'appel des différents noyaux par le processeur central.

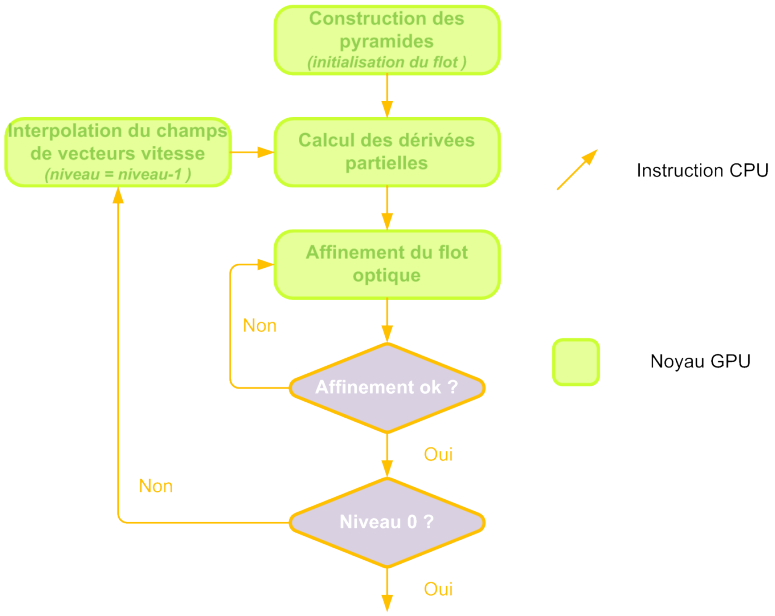


FIG. 7.5 – Diagramme fonctionnel de l'estimation du flot optique sur GPU.

La gestion des entrées-sorties est identique pour tous les *kernels*. Les données entrantes, qu'il s'agisse d'images, de dérivées partielles ou encore de champ de vecteurs vitesse, sont copiées dans le cache de texture pour en accélérer l'accès. Un autre atout des unités de textures réside dans leur capacité à produire l'interpolation bilinéaire en tout point d'une texture, non pas de façon logicielle mais matérielle. Ainsi, l'interpolation bilinéaire du champ de déplacement, d'un niveau de la pyramide à l'autre, n'augmente en rien le temps de calcul sur GPU, contrairement à sa version CPU. Le résultat de chaque *kernel* est copié dans la mémoire

globale de la carte graphique, seul module disponible en écriture dont la durée de vie des données ne dépend pas de celle du noyau.

L'utilisation d'un *profiler* montre que 87% du temps GPU, correspondant à l'exécution de l'ensemble du processus, est dédié au noyau du calcul du flot optique. On concentre donc l'évaluation de l'implémentation CUDA sur ce *kernel*, avec l'étude de deux grandeurs significatives : la bande passante et le nombre d'opérations flottantes effectuées par seconde. La bande passante d'un noyau, par abus de langage, désigne le débit d'information moyen et permet de contrôler l'optimisation des accès mémoire. Ces derniers se révèlent d'autant plus performants que l'on respecte certaines règles communes à la majeure partie des architectures vectorielles, telles que l'alignement des données lues ou écrites. L'interface GPU-GPU des circuits NVIDIA G80, offre une bande passante maximale de $80Go/s$ [98], dont $60Go/s$ sont exploitées par l'implémentation proposée. Il s'agit d'une valeur standard pour un *kernel* dont l'intensité arithmétique est plutôt faible, avec environ $40Gflops$[4].

L'exécution complète de l'algorithme d'estimation du flot optique, pour une paire d'images de 640×480 pixels, 4 niveaux de pyramide et une fenêtre d'ouverture Ω_{ROI} de 10×10 pixels, nécessite 65 millisecondes sur une carte graphique, contre 5 secondes environ pour un processeur *mono-core* de $3.2Ghz$, soit un gain d'un facteur 80. L'implémentation est disponible en ligne sous la forme d'une librairie CUDA, sur le site de la firme NVIDIA, à l'adresse `http://www.nvidia.com/object/cuda_home.html`.

Tensor Voting

Tout comme l'estimation du flot optique, le *Tensor Voting* est hautement parallélisable, puisque chaque élément de l'espace considéré peut être traité indépendemment. Il se prête donc parfaitement à l'exploitation de processeurs vectoriels tels que les GPU modernes.

Le *Tensor Voting* peut être divisé en deux processus : d'une part le *vote*, au cours duquel chaque tenseur accumule l'information induite par l'agencement de son voisinage et, d'autre part, la *décomposition* des tenseurs résultants en valeurs et vecteurs propres, de manière à caractériser géométriquement l'ensemble des n éléments de l'espace étudié. Lorsque ce dernier est de dimension 4, chaque tenseur est représenté par une matrice symétrique de taille 4×4 dont seules 10 entrées

[4] Résultats obtenus pour l'exécution du noyau sur un niveau de pyramide.

suffisent à le définir. Pour ne pas avoir à stocker temporairement sur la mémoire globale, les $n \times 10$ valeurs nécessaires à la sauvegarde du champs tensoriel entre le vote et la décomposition des tenseurs, on opte pour une solution regroupant les deux processus au sein d'un même noyau. La décomposition en valeurs propres est issue de l'implémentation présentée dans *numerical recipes* : les boucles sont déroulées pour développer certains calculs et limiter ainsi l'utilisation des ressources GPU.

L'exécution du *Tensor Voting*, à l'aide d'un processeur *mono-core* cadencé à $3.2Ghz$, nécessite une vingtaine de secondes pour une image de 640×480 pixels et un paramètre de diffusion σ égal à 4. Il suffit en revanche de 25 millisecondes sur un GPU de type G80 pour effectuer la même tâche, permettant ainsi l'emploi du *Tensor Voting* au sein d'applications temps-réel.

7.2.3 Perspectives

Les résultats présentés correspondent aux mesures réalisées à l'aide d'une carte NVIDIA Tesla C870, de type G80. Cette dernière comprend 128 processeurs de flux et une interface mémoire de 384 bits, pour une bande-passante atteignant $76.8Go/s$. Afin d'améliorer encore les temps de calcul énoncés, une première solution consiste à employer un circuit graphique plus performant. La gamme NVIDIA GT200, par exemple, permet de doubler le nombres d'unités de calcul ainsi que la mémoire allouée à chacun d'eux. Aucune modification du code n'est pour autant nécessaire afin de bénéficier du surcroît de ressources disponible, la division par blocs permettant de répartir équitablement, et de façon transparente pour l'utilisateur, la charge de calcul sur l'ensemble des processeurs de flux.

Une seconde solution repose sur l'utilisation simultanée de plusieurs GPU, et nécessite donc de diviser manuellement le traitement des images entre diffèrents circuits graphiques. Dans le cas d'algorithmes parallélisables jusqu'au niveau pixelique, à savoir pour lesquels chaque point est traité indépendamment de ses voisins, les images peuvent être divisées à parts égales entre les différents GPU. Le code CUDA reste alors inchangé puisque seules les dimensions, de l'image traitée par chacun des circuits graphiques, diffèrent de l'exécution originale mono-GPU. Par ailleurs, tandis qu'ATI propose d'ores et déjà un pilote Stream capable de gérer plusieurs cartes comme une seule, NVIDIA devrait prochainement faire de même.

7.3 Conclusions de la thèse

Contexte et solutions

On trouve à l'origine des travaux menés dans le cadre de cette thèse, l'importance grandissante apportées aux techniques de perception reposant sur des méthodes d'apprentissage, essentiellement supervisées. Sans discuter leur place prééminente dans le processus de perception humain, que l'on tente souvent de reproduire, il est cependant essentiel de comprendre qu'elles ne sont pas suffisantes dans un milieu ouvert, sans éliminer toute notion d'inconnu (variation du revêtement au sol, nouveau type d'obstacle, etc.). Il faut donc pouvoir palier l'incomplétude d'une base d'apprentissage afin de garantir un niveau minimum de fiabilité. Les résultats présentés dans ce manuscrit répondent à cette problématique par le biais d'une approche locale, sans connaissance *a priori* de la scène observée.

En dehors des méthodes, dites, par apprentissage, la perception monoculaire de l'espace libre repose le plus souvent sur des approches éparses ou identifiant le sol à l'aide de critères photométriques divers. Les méthodes basées sur la recherche de marquage au sol, ainsi que celles fondées sur l'étude colorimétrique ou texturelle de l'espace navigable, sont les plus courantes. Dans le meilleur cas, toutefois, la première solution n'offre qu'un modèle surfacique de la route sans information sur les obstacles qui s'y trouvent, tandis que la seconde n'est pas suffisante pour définir directement la géométrie du terrain. Enfin, la combinaison des deux restreint la fonctionnalité du système perceptif aux environnements structurés, munis d'un revêtement uniforme et d'un marquage signalétique au sol.

L'étude, proposée au chapitre 6, des contraintes épipolaires liées au mouvement image induit par l'*ego-motion*, permet de déterminer la géométrie de l'espace navigable, en dehors de toute considération photométrique associée à la nature du terrain rencontré. Ces contraintes sont calculées en tout point de l'image pour assurer la classification de chaque élément observé dans la scène. Elle permettent également, de manière analogue au système perceptif humain, qui ne perçoit le relief par stéréoscopie qu'en deçà de la distance utile pour conduire, d'estimer la profondeur des obstacles, dans le repère caméra, uniquement d'après l'étude des mouvements parallaxes.

Finalement, tandis que la plupart des travaux de perception visuelle en robotique mobile différencient la détection de l'espace navigable et des obstacles en mouvement, on propose une solution unifiée, aisément intégrable dans un module de perception plus haut niveau. La figure 7.6 résume l'ordonnancement des diffé-

rentes étapes de cette solution :

1. Les points de deux images acquises consécutivement sont appariés pour établir le flot optique. Deux champ de vecteurs sont ainsi obtenus : l'un, sans initialiser l'algorithme pyramidal de Lucas et Kanade, et l'autre, à partir du déplacement calculé pour la paire d'images précédente.

2. La combinaison des deux champs de déplacement, à l'aide du *Tensor Voting* 4-D, permet d'obtenir une estimation du mouvement apparent plus fine, notamment aux bords de l'image et à proximité des régions pour lesquelles des points sont occultés dans l'image courante.

3. Une seconde passe de *Tensor Voting* 4-D, sur le flot optique combiné, fournit une évaluation du champs de déplacement final. On supprime ainsi les estimations inhomogènes avec leur voisinage.

4. A partir du flot optique ainsi filtré, on calcule ensuite, de manière robuste, le modèle homographique induit par l'espace navigable, supposé localement plan. La distance entre ce modèle et le déplacement estimé permet d'identifier l'espace libre pour circuler. De façon duale, les obstacles (statique et mobiles) sont également identifiés. La décomposition de l'homographie du plan au sol renseigne également sur le déplacement de la caméra entre les acquisitions. L'homographie correspondant à la transformation IPM du sol est calculée.

5. Le mouvement apparent est ensuite décomposé pour obtenir sa composante parallaxe, grâce à laquelle il est possible d'estimer la profondeur relative des points de l'image correspondant aux éléments statiques de l'espace. En prenant comme référence les points de l'espace navigable dont on connaît la position dans $\mathbb{R}^3$, on calcule alors la profondeur réelle associée à chaque pixel de l'image.

6. Pour terminer, les discontinuités du flot optique, identifiées à l'aide du *Tensor Voting*, et la carte des profondeurs précédemment calculée permettent de construire une structure hiérarchique analogue au graphe de scène utilisé en infographie. En comparant la profondeur de chaque élément connexe à l'espace navigable avec la distance du centre optique jusqu'au point de contact de cet élément avec le sol, on identifie enfin les obstacles mobiles.

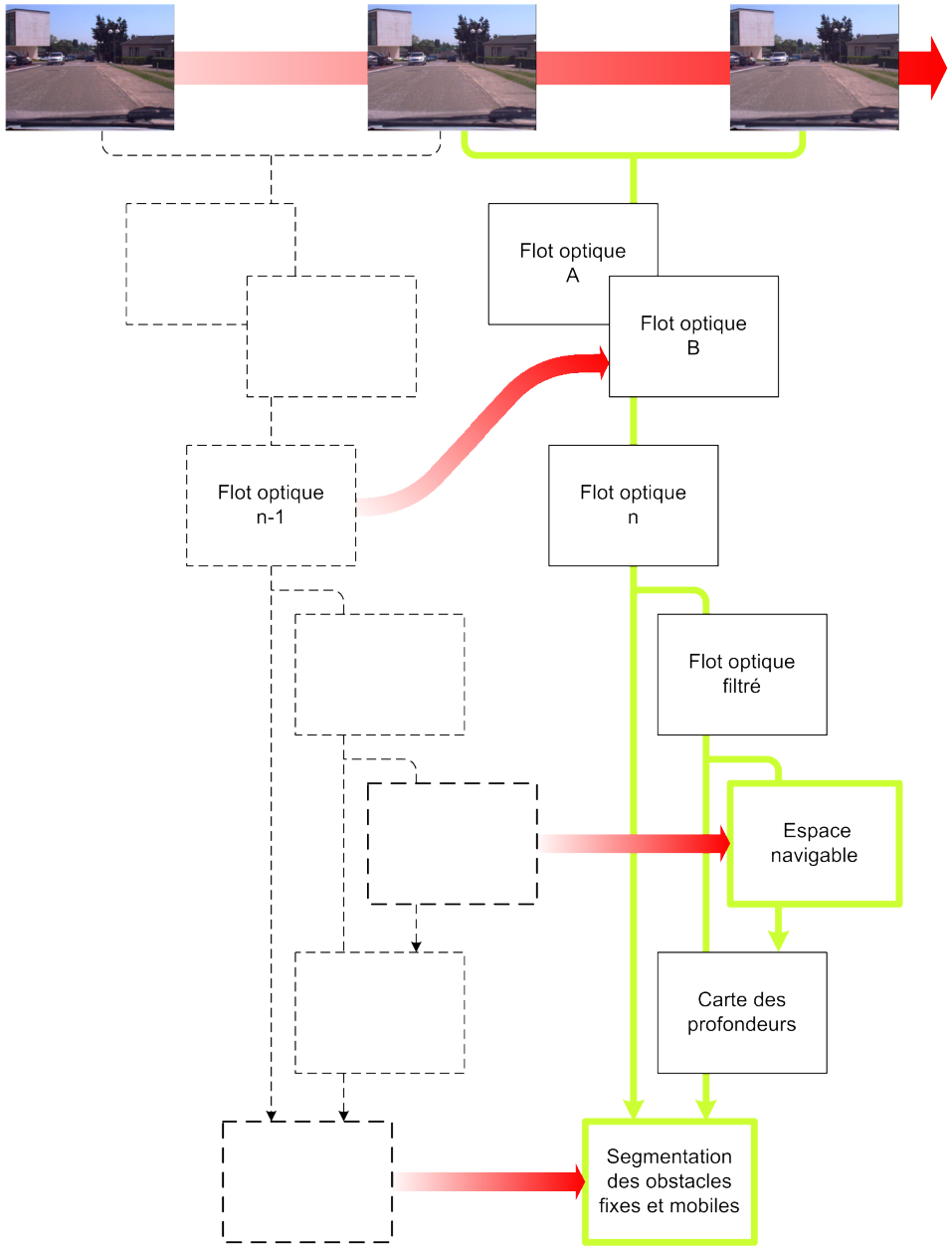


FIG. 7.6 – Diagramme fonctionnel de l'approche proposée pour la perception monoculaire de l'environnement.

Apports et ouverture

Les contributions de ce travail sont multiples. La formulation du processus de perception visuelle pour la conduite automatisée, dans sa globalité, a ainsi permis de proposer une solution cohérente pour identifier, selon des critères topographiques et dynamiques, les différents éléments de la scènes observée. La nécessité d'une telle approche a été établi, pour pouvoir classifier sémantiquement chaque objet de l'environnent.

Si définir la géométrie de $\mathbb{R}^3$ à l'aide du mouvement apparent n'est pas récent, les recherches en perception pour la robotique mobile se limitent généralement à une étude non dense de la scène afin de satisfaire la contrainte de temps-réel. Les approches résultantes reposent donc, le plus souvent, sur l'emploi de points d'intérêt à partir desquels elles sont en mesure d'estimer un modèle global de certains éléments de la scène. Dans le cas du plan de l'espace navigable, différents travaux ont, par le passé, déjà proposé d'estimer le modèle homographique induit, à l'aide de plusieurs points caractéristiques mis en correspondance sur deux images successives. L'homographie servait alors à transformer la première image, de sorte qu'au niveau du sol, la nouvelle image soit identique à la seconde. Il s'agissait finalement, à partir d'une information éparse, de pouvoir vérifier de manière dense, la consistance du mouvement apparent des points de l'image et du modèle homographique. Toutefois la figure 5.7 du chapitre 5 montre combien l'information obtenue est alors difficilement interprétable.
A l'inverse, en estimant le mouvement apparent en tout point de l'image, il est possible de comparer directement le déplacement homographique avec le flot optique et ainsi discriminer les points de l'espace navigable. Pour valider cette approche dans le contexte des véhicules automatisés, l'algorithme pyramidale de Lucas et Kanade a été porté sur GPU, afin de démontrer la possibilité d'estimer le mouvement image en temps réel. Cette implémentation, disponible en ligne sur le portail NVIDIA CUDA, est régulièrement téléchargée par divers membre de la communauté scientifique, attestant ainsi de l'utilité de ce développement.

L'appariement des points de deux images, pour l'estimation dense du flot optique, peut être bruité. Aussi, propose-ton d'employer le formalisme conçu par G. Medioni, le *Tensor Voting*, afin d'améliorer l'estimation du mouvement apparent puis de l'évaluer et, enfin, de le segmenter.
Comme l'explique le chapitre 3, pour l'estimation du flot optique, l'approche pyramidale et le raffinement temporel sont deux procédés difficilement compatibles :

aux niveaux les plus élevés de la pyramide, l'ouverture utilisée pour estimer le flot optique est trop importante par rapport à la norme du déplacement précédemment estimé et sous-échantillonné, pour que ce dernier influe significativement sur la convergence du processus. En combinant grâce au *Tensor Voting,* le résultat de l'approche pyramidale avec avec un second champ de déplacement, calculé sur un niveau et initialisé à l'aide du flot optique précédent, on propose cependant d'associer efficacement l'hypothèse de continuité temporelle du mouvement apparent et la robustesse d'une méthode multi-résolutions.

Par la suite, l'évaluation du flot optique final, selon un critère de continuité dans l'espace (x, y, v_x, v_y), optimise le choix des points utilisés lors du calcul de l'homographique induite par le plan du sol. On réduit ainsi le nombre d'itérations nécessaires à l'algorithme d'estimation robuste RANSAC pour trouver la solution. En outre, l'image des discontinuités résultant du *Tensor Voting* sert également de support à la segmentation du mouvement apparent par *watershed*.

Enfin, comme pour l'estimation du flot optique, une implémentation temps-réel du *Tensor Voting* a été réalisée, grâce à l'emploi de GPU, via l'interface de programmation CUDA.

L'identification des parties mobiles de la scène est couramment assurée par l'utilisation de contraintes épipolaires sur le mouvement résiduel parallaxe des points de l'image. Ces contraintes utilisent, explicitement ou non, la position de l'épipôle dans l'image courante. L'axe du déplacement parallaxe des objets statiques converge vers cet épipôle, et l'on définit alors tout objet ne satisfaisant pas cette contrainte comme un obstacle mobile. Cependant, lorsque les obstacles se déplacent le long de l'axe optique tandis que le véhicule avance en ligne droite, leur mouvement résiduel est confondu avec l'approximation du déplacement parallaxe des éléments fixes. On présente donc un critère sur la norme du déplacement parallaxe, à travers l'estimation de la profondeur des points de l'image connexes au plan du sol.

Le processus de perception présenté dans ce document repose donc sur l'analyse dense de chaque paire d'acquisitions, par l'étude de contraintes géométriques et dynamiques liées à la projection de l'environnement dans le plan focal. Il peut être employé seul, ou intégré dans un système perceptif plus complet, couplé par exemple à un module de reconnaissance par apprentissage. Dans ce dernier cas, le module de perception par apprentissage pourrait être suppléer si un obstacle non répertorié se présentait. L'orientation des travaux à venir devra tendre vers

cet objectif afin de développer un module de perception fiable et performant. Plus spécifiquement, il sera intéressant d'améliorer la détection de l'espace navigable à longue distance en s'inspirant des récents travaux de Y. Lecun [99] en stéréovision. Le principe consistera à déterminer, géométriquement et à proximité du véhicule, l'espace navigable, de manière à utiliser l'information texturelle de ce dernier pour étendre sa détection, jusqu'à l'horizon, par apprentissage.

Notations

1. Espaces linéaires

$\mathbb{R}^3$	Espace linéaire des réels 3-D
$\mathbb{E}^3$	Espace euclidien 3-D
$\mathbb{P}^2$	Espace projectif 2-D

2. Notations vectorielles et matricielles

a	Scalaire
$\mathbf{v}$	Vecteur
$\hat{\mathbf{v}}$	Vecteur unitaire
$\mathbf{v}_\perp$	Un vecteur orthogonal à $\mathbf{v}$
$\mathtt{M}$	Matrice

3. Primitives géométriques

P	Un point générique de $\mathbb{R}^3$
p	Un point générique de $\mathbb{P}^2$
$\mathbf{X}$	Les coordonnées $(X,Y,Z)^T \in \mathbb{R}^3$ d'un point P de l'espace
$\widetilde{\mathbf{X}}$	Représentation homogène de $\mathbf{X}$, avec $\widetilde{\mathbf{X}} = (X,Y,Z,1)^T \in \mathbb{R}^4$
$\mathbf{x}$	Les coordonnées $(x,y)^T \in \mathbb{R}^2$ d'un point p du plan focal
$\widetilde{\mathbf{x}}$	Représentation homogène de $\mathbf{x}$, avec $\widetilde{\mathbf{x}} = (x,y,z,1)^T \in \mathbb{R}^3$

Bibliographie

[1] J.-Y. Bouguet, *"Pyramidal implementation of the Lucas Kanade feature tracker : Description of the algorithm",* in Intel Research Laboratory, Technical Report (1999).

[2] T. Fraichard and H. Asama, *"Inevitable collision states. A step towards safer robots ?"*, in Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems (2003), pp. 388-393.

[3] S. Petti and T. Fraichard, *"Safe navigation of a car-like robot in a dynamic environment"*, in Proceedings of the European Conference on Mobile Robots (2005).

[4] Y. Ma, S. Soatto, J. Kosecká and S. Shankar Sastry, *"An invitation to 3-D vision : from images to geometric models"*, in Interdisciplanary applied mathematics (2000), Vol. 26, Springer ed..

[5] J. E. Cutting and P. M. Vishton The, *"Perceiving layout and knowing distances : the integration, relative potency, and contextual use of different information about depth"*, in Perception of Space and Motion (1995), pp. 69–117, Academic Press.

[6] I. P. Howard, and B.J. Rogers, *"Binocular Vision and Stereopsis"*, in Oxford University Press (1995).

[7] B. Rogers, and M. Graham, *"Motion parallax as an independent cue for depth perception"*, in Perception (1979), No 8, pp. 125–134.

[8] J. Zhou and B. Li, *"Robust ground plane detection with normalized homography in monocular sequences from a robot platform"*, in Proceedings of the IEEE International Conference on Image Processing (2006).

[9] A. Wedel, T. Schoenemann, T. Brox and D. Cremers, *"WarpCut - Fast obstacle segmentation in monocular video"*, in Lecture Notes in Computer Science (2007), pp. 264-273, Springer ed..

[10] W. Trobin, T. Pock, D. Cremers and H. Bischof, *"An unbiased second-order prior for high accuracy motion estimation"*, in DAGM Symposium (2008), pp. 396-405.

[11] D. Aubert and C. Thorpe, *"Color image processing for navigation : two road trackers"*, in CMU Robotics Institute Laboratory, Technical Report (1990).

[12] S. Beucher and M. Bilodeau, *"Road segmentation and obstacle detection by a fast watershed transformation"*, in Proceedings of the IEEE Intelligent Vehicles Symposium (1994), pp. 296-301.

[13] A. Broggi and S. Berte, *"Vision-based road detection in automotive systems : a real-time expectation driven approach"*, in Journal of Artificial Intelligence Research (1995), vol. 3, pp. 325-348.

[14] F. Paetzold and U. Franke, *"Road recognition in urban environment"*, in Image and Vision Computing (2000), vol. 18, pp. 377-387.

[15] T. M. Huang, V. Kecman and I. Kopriva, *"Kernel based algorithms for mining huge data sets, supervised, semi-supervised, and unsupervised learning"*, in Springer-Verlag (2006).

[16] A. Khammari, F. Nashashibi, Y. Abramson and C. Laurgeau, *"Vehicle detection combining gradient analysis and AdaBoost classification"*, in Proceedings of the IEEE International Conference on Intelligent Transportation Systems (2005).

[17] M. A. Sotelo, I. Parra, D. Fernández and E. Naranjo, *"Pedestrian detection using SVM and multi-feature combination"*, in Proceedings of the IEEE International Conference on Intelligent Transportation Systems (2006).

[18] D. Hoiem, A. A. Efros and M. Hebert, *"Closing the loop in scene interpretation"*, in Proceedings of the IEEE computer society conference on Computer Vision and Pattern Recognition (2008).

[19] N. Soquet and D. Aubert, *"Road Segmentation Supervised by an Extended V-Disparity Algorithm for Autonomous Navigation"*, in Proceedings of the IEEE Intelligent Vehicles symposium (2007).

[20] N. Simond and M. Parent, *"Obstacle detection from IPM and Super-Homography"*, in Proceedings of the IEEE International Conference on Intelligent Robots and Systems (2007).

[21] M. Yang, Q. Yu, H. Wang and B. Zhang, *"Vision-based real-time obstacle detection and tracking for autonomous vehicle guidance"*, in. SPIE Real-Time Imaging VI, Vol 4666, pp. 65-74, Nasser Kehtarnavaz Ed.

[22] W. Miled, J.-C. Pesquet and M. Parent, *"Robust obstacle detection based on dense disparity maps"*, in the Eleventh International Conference on Computer Aided Systems Theory (2007).

[23] S. Lefebvre, S. Ambellouis and F. Cabestaing, *"Obstacles detection on a road by dense stereovision with 1D correlation windows and fuzzy filtering"*, in Proceedings of the IEEE International Conference on Intelligent Transportation Systems (2006).

[24] N. B. Touzene, and S. Larabi, *"Obstacle detection from uncalibrated cameras"*, in Proceedings of the IEEE Panhellenic Conference on Informatics (2008).

[25] R. Labayrade, and D. Aubert, *"In-vehicle obstacles detection and characterization by stereovision"*, in Proceedings of the IEEE International Conference on Cognitive Computer Vision Systems (2003).

[26] J. Wang , Z. Hu, H. Lu and K. Uchimura, *"Motion detection in driving environment using U-V-disparity"*, in Lecture Notes in Computer Science (2006), pp. 307-316, Springer : Berlin.

[27] Y. Gao, *"Etude psychophysiologique de la vision en relief humaine en télévision stéréo"*, Thèse de Génie Biologique et Médical à l'institut national des sciences appliquées de Lyon (1992).

[28] H. H. Nagel, *"Extending oriented smoothness constraint into the temporal domain and the estimation of derivatives of optical flow"*, in Lecture Notes in Computer Science (1990), pp. 139-148, Springer : Berlin.

[29] M. J. Black, *"Robust incremental optical flow"*, in Ph.D. thesis (1992), Yale University.

[30] C. Koch, H. T. Wang, B. Mather, A. Hsu and H. Suarez, *"Computing optical flow in resistive networks and in the primate visual system"*, in Proceedings of the IEEE Workshop on Visual Motion (1989), pp. 62-72.

[31] H. H. Nagel, *"On the estimation of optical flow relations between different approaches and some new results"*, in Articial Intelligence (1987), No. 33, pp. 299-324.

[32] A. D. Jepson and M. Black, *"Mixture models for optical flow computation."*, in IEEE Computer Society. Conference on Computer Vision and Pattern Recognition (1993), pp. 760-761.

[33] A. Singh, *"Incremental estimation of image flow using Kalman filter"*, in Proceedings of the IEEE Workshop on Visual Motion (1991), pp. 36-43.

[34] B. K. P. Horn and B. G. Schunck, *"Determining optical flow"*, in Artificial Intelligence (1981), No. 17, pp. 185-204.

[35] J. L. Barron, D. J. Fleet and S. S. Beauchemin, *"Performance of Optical Flow Techniques"*, in International Journal of Computer Vision (1994), Vol. 12, No. 1, pp. 43-77, Springer : US.

[36] B. Lucas and T. Kanade, *"An iterative image registration technique with an application to stereovision"*, in Proceedings of the DARPA IU Workshop (1981), pp. 121-130.

[37] S. Beauchemin and J. Barron, *"The computation of optical flow"*, in ACM Computing Surveys (1995), Vol. 27, No. 3, pp. 433–467.

[38] D. Pellerin, A. Spinéi and A. Guérin-Dugué, *"Calcul du flot optique par filtrages de Gabor combinés"*, in Traitement du Signal (1996), Vol. 13, No. 1.

[39] D. J. Heeger, *"Optical flow using spatio-temporal filters"*, in International Journal of Computer Vision (1988), Vol. 1, pp. 279-302.

[40] V. Argyriou and T. Vlachos, *"A study of sub-pixel motion estimation using phase correlation"*, in Technical Report (2004), University of Surrey.

[41] C. D. Kuglin and D. C. Hines, *"The phase correlation image alignment method"*, in Proceedings of the Conference Cybernetics Society (1975), pp. 163-165.

[42] Y. T. Wu, T. Kanade, J. Cohn and C. C. Li, *"Optical flow estimation using wavelet motion model"*, in Proceedings of the Sixth International Conference on Computer Vision (1998).

[43] R. Szeliski and H.-Y. Shum, *"Motion Estimation with Quadtree Splines"*, in Technical Report (1995), Cambridge Research Lab.

[44] C. Bernard, *"Discrete Wavelet Analysis for Fast Optic Flow Computation"*, in Applied and Computational Harmonic Analysis (2001), Vol. 11, No. 1, juillet 2001, pp. 32-63.

[45] W. Li and E. Salari, *"Successive elimination algorithm for motion estimation"*, in IEEE Transactions on Image Processing (1995), Vol. 4, No. 1.

[46] X. Jing and L. P. Chaud, *"An efficient three-step search algorithm for block motion estimation"*, in IEEE Transactions on Multimedia (2004), Vol. 6, No. 3.

[47] Y. Nie and K. K. Ma, *"Adaptive rood pattern search for fast block matching motion estimation"*, in IEEE Transactions on Image Processing (2002), Vol. 11, No. 12.

[48] F. Essannouni, R. O. H. Thami, A. Salam and D. Aboutajdine, *"An efficient fast full search block matching algorithm using FFT algorithms"*, in International Journal of Computer Science and Network Security (2006), Vol. 6, No. 3b.

[49] *"H.264 : Advanced video coding for generic audiovisual services"*, in Technical Report (2003), International Telecommunication Union, `http://www.itu.int/rec/T-REC-H.264-200305-S`

[50] C. Q. Davis, Z. Z. Karul, D. M. Freeman, *"Equivalence of subpixel motion estimators based on optical flow and block matching"*, in International Symposium on Computer Vision (1995).

[51] J. Barron and R. Klette, *"Quantitative Color Optical Flow"*, in 16th International Conference on Pattern Recognition (2002), Vol. 4, pp. 251–255.

[52] P. Golland and A. M. Bruckstein, *"Motion from color"*, in Technical Report #9513 (1997), Computer Science Department, Technion.

[53] J. Marzat, *"Estimation temps réel du flot optique"*, in Rapport de Stage Ingénieur (2008), Institut National de Recherche en Informatique et Automatique (2008).

[54] J. Marzat, Y. Dumortier and A. Ducrot, *"Real-time Dense and Accurate Parallel Optical Flow using CUDA"*, International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2009).

[55] E. Mémin, *"Estimation du flot-optique : contributions et panorama de différentes approches"*, in Habilitation à Diriger des Recherches (2003), Université Rennes 1, Institut de Formation Supérieure en Informatique et en Communication.

[56] M. Orkisz and P. Clarysse, *"Estimation du flot optique en présence de discontinuités : une revue"*, in Traitement du Signal (1996), Vol. 13(5), pp 489-513.

[57] R. Deriche and O. Faugeras, *"Les EDP en traitement des images et vision par ordinateur"*, in Rapport de Recherche (1995), Institut National de Recherche en Informatique et Automatique.

[58] D. A. Danielson, *"Vectors and tensors in engineering and physics"*, second edition (2003), in Advanced Book Program, Westview Press.

[59] G. Medioni and M. S. Lee, *"A computational framework for segmentaion and grouping"*, (2000), Elsevier edition.

[60] M. Nicolescu and G. Medioni, *"4-D voting for matching, densification and segmentation into motion layers"*, in International Conference on Pattern Recognition (2002), Vol. 3, pp. 303-308.

[61] Y. Dumortier, I. Herlin and A. Ducrot, *"4-D Tensor Voting motion segmentation for obstacle detection in autonomous guided vehicle"*, in IEEE Intelligent Vehicles Symposium (2008), pp. 379-384.

[62] Y. Dumortier, I. Herlin, *"Monocular Moving Obstacle Detection for Autonomous Guided Vehicles"*, in Transport Research Arena, Young European Arena of Research (2008).

[63] K. Mullet and D. Sano, *"Designing visual interfaces : Communication oriented techniques"*, (1995), Englewood Cliffs, NJ : Prentice Hall.

[64] B. Giai-Checa, P. Bouthemy and T. Vieville, *"Segment-based detection of moving objects in a sequence of images"*, in International Conference on Pattern Recognition (1994), pp. 384-389.

[65] G. Csurka and O. Faugeras, *"Algebraic and geometrical tools to compute projective and permutation invariants"*, in IEEE Trans. Pattern Analysis and Machine Intelligence (1999), Vol. 21, pp. 58-65.

[66] C. Cappelle, M. El Badaoui El Najjar, F. Charpillet and D. Pomorski, *"Obstacle detection and localization method based on 3D model : Distance validation with ladar"*, in IEEE International Conference on Robotics and Automation (2008), pp. 4031-4036.

[67] Z. Zhang, M. Li, K. Huang and T. Tan, *"3D Model based Vehicle Localization by Optimizing Local Gradient based Fitness Evaluation"*, int IEEE International Conference on Pattern Recognition (2008), pp. 1-4.

[68] A. Broggi and S. Berte, *"Vision-based road detection in automotive systems : a real-time expectation driven approach"*, in Journal of Artificial Intelligence Research (1995), Vol. 3, pp. 325-348.

[69] Y. Li, K. He and P. J. Tsinghua, *"Road markers recognition based on shape information"*, in IEEE Intelligent Vehicles Symposium (2007), pp 117-122.

[70] A. Kuehnle, *"Symmetry-based recognition of vehicle rears"*, in Pattern Recognition Letters (1991), pp. 249-258.

[71] T. Zielke, M. Brauckmann and W. von Seelen, *"Intensity and edge-based symmetry detection with an application to car-following"*, in Graphical Models and Image Processing (1993), No. 2, pp. 177-190.

[72] Y. He, H. Wang and B. Zhang, *"Color-based road detection in urban trafic scenes"*, in IEEE Intelligent Transportation Systems (2004), Vol. 5, pp. 309-318.

[73] P. Paalanen, V. Kyrki and J.-K. Kamarainen, *"Towards monocular on-line 3D reconstruction"*, in Workshop on Vision in Action : Efficient strategies for cognitive agents in complex environments (2008).

[74] T.L. Gandhi, S. Devadiga, R. Kasturi and O.I. Camps, *"Detection of obstacles on runway using ego-motion compensation and tracking of significant features"*, in Proceedings or the 3rd IEEE Workshop on Applications of Computer Vision (1996), pp. 168-173.

[75] S. Beucher and F. Meyer, *"The morphological approach to segmentation : the watershed transformation"*, in Mathematical Morphology in Image Processing (1993), Ed. E.R. Dougherty, pp. 433-481.

[76] L. Najman, M. Couprie and G. Bertrand, *"Watersheds, mosaics, and the emergence paradigm"*, in Discrete Applied Mathematics (2005), Vol. 147, No 2-3, pp. 301-324 .

[77] J.C. Klein, *"Conception et réalisation d'une unité logique pour l'analyse quantitative d'images"*, in Ph.D. thesis (1976), Université de Nancy.

[78] J. Serra and P. Salembier, *"Connected operators and pyramids"*, in Image Algebra and Mathematical Morphology (1993), Vol. 2030, pp. 65-76, ed. SPIE.

[79] P. Salembier and J. Serra, *"Flat zones filtering, connected operators and filters by reconstruction"*, in IEEE Transactions on Image Processing (1995), Vol. 3, No. 8, pp. 1153-1160.

[80] L. Vincent, *"Morphological gray scale reconstruction in image analysis applications and efficients algorithms"*, in IEEE Transactions on Image Processing (1993), Vol. 2, No. 2, pp. 176-201.

[81] F. Meyer and S. Beucher, *"Morphological segmentation"*, in Journal of Visual Communication and Image Representation (1990), Vol. 1, No. 1, pp. 21-46.

[82] P. Salembier, L. Torres, F. Meyer and C. Gu, *"Region-based video coding using mathematical morphology"*, in Proceedings of IEEE (papier invité) (1995), Vol 83, No. 6, pp. 843-857.

[83] C. Vachier, *"Utilisation d'un critère volumique pour le filtrage d'image"*, in 11th Conference on Shape recognition and artificial intelligence (1998), pp. 307-315.

[84] J.-Y. Bouguet, *"Camera Calibration Toolbox for Matlab"*, `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[85] Z. Zhang, *"A flexible new technique for camera calibration"*, in IEEE Transactions on Pattern Analysis and Machine Intelligence (2000), 22(11), pp. 1330-1334.

[86] Z. Zhang, *"Flexible Camera calibration by viewing a plane from unknown orientations"*, in International Conference on Computer Vision (1999), pp. 666-673.

[87] L. S. Davis, D. Oberkampf and D. Dementhon, *"Iterative pose estimation using coplanar feature points"*, in Computer Vision and Image Understanding (1996), 63(3), pp. 495-511.

[88] D. DeMenthon and L. S.Davis, *"Model-based object pose in 25 lines of code"*, in European Conference on Computer Vision (1992), pp. 335-343.

[89] Y. Dumortier, M. Kais and R. Benenson, *"Real-time vehicle motion estimation using texture learning and monocular vision"*, in International Conference on Computer Vision and Graphics (2006).

[90] O. Faugeras and F. Lustman, *"Motion and structure from motion in a piecewise planar environment"*, in International Journal of Pattern Recognition and Artificial Intelligence (1988), 2(3), pp. 485–508.

[91] O. Faugeras, *"Three-dimensional computer vision : a geometric viewpoint"*, in MIT Press (1993).

[92] Z. Zhang, and A.R. Hanson, *"3D Reconstruction based on homography mapping"*, in Proceedings of ARPA (1996), pp. 1007-1012.

[93] A. Agarwal, C. V. Jawahar and P. J. Narayanan, *"A survey of planar homography estimation techniques"*, in technical report of Indian International Institute of Information Technology (2005).

[94] E. Malis and M. Vargas, *"Deeper understanding of the homography decomposition for vision-based control"*, in Rapport de Recherche de l'INRIA (2007).

[95] M. Irani and P. Anadan, *"Parallax geometry of pairs of points for 3-D scene analysis"*, in European Conference on Computer Vision (1996), pp. 17-30.

[96] L. Seiler, et al., *"Larrabee : a many-core x86 architecture for visual computing"*, in ACM Transaction on Graphics (2008), Vol. 27, No. 3, pp. 1-15.

[97] *"NVIDIA CUDA : Compute Unified Device Architecture. Programming guide"*, in NVIDIA technical report (2008).

[98] T. R. Halfhill, *"Parallel Processing with CUDA"*, in Microprocessor Report (2008).

[99] R. Hadsell, P. Sermanet, M. Scoffier, A. Erkan, K. Kavackuoglu, U. Muller and Y. LeCun, *"Learning long-range vision for autonomous off-road driving"*, in Journal of Field Robotics (2009), Vol. 2, No. 26, pp. 120-144.

[100] R. Hartley and A. Zisserman, *"Multiple view geometry in computer vision"*, in Cambridge University Press (2000).

[101] G. W. Stewart, *"On the early history of the singular value decomposition"*, in technical report of University of Maryland (1992).